Hans-Otto Günther
Kap-Hwan Kim
Herbert Kopfer (Eds.)

LOGMS

The 2012 International Conference on Logistics and Maritime Systems

Proceedings

University of Bremen • Germany
August 22-24 • 2012

Universität Bremen
KIESERLING STIFTUNG
wiwib Wirtschaft-Wissenschaft-Bremen

Editors

Hans-Otto Günther

Technical University of Berlin, Department of Production Management
Straße des 17. Juni 135, 10623 Berlin, Germany
E-mail: hans-otto.guenther@tu-berlin.de

Kap-Hwan Kim

Pusan National University, Department of Industrial Engineering
Jangjeon-dong, Gumjeong-gu, Busan 609-735, Korea
E-mail: kapkim@pusan.ac.kr

Herbert Kopfer

University of Bremen, Chair of Logistics
Wilhelm-Herbst-Str. 5, 28359 Bremen, Germany
E-mail: kopfer@uni-bremen.de

Table of Contents

A MATHEURISTIC FOR THE LINER SHIPPING NETWORK DESIGN PROBLEM

Berit Dangaard Brouer[1] and Guy Desaulniers[2]

[1]*DTU Management Engineering, Technical University of Denmark, Produktionstorvet, 2800 Lyngby, Denmark, blof@dtu.dk*

[2]*GERAD, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Canada, guy.desaulniers@gerad.ca*

ABSTRACT

We present a matheuristic, an integer programming based heuristic, for the Liner Shipping Network Design Problem. The heuristic applies a greedy construction heuristic based on an interpretation of the liner shipping network design problem as a multiple quadratic knapsack problem. The construction heuristic is combined with an improvement heuristic with a neighborhood defined by the solution space of a mixed integer program. The mixed integer program optimizes the removal and insertion of several port calls on a liner shipping service. The objective function is based on evaluation functions for revenue and transshipment of cargo along with in/decrease of vessel- and operational cost for the current solution. The evaluation functions may be used by heuristics in general to evaluate changes to a network design without solving a large scale multicommodity flow problem.

Keywords: liner shipping, matheuristic, network design, mathematical programming

1 INTRODUCTION

Liner shipping is the mass transit system of the ocean ways with regular scheduled services of varying capacity between geographical regions. Liner shipping and containerized transportation of goods over sea is a key component in todays supply chains. Approximately 400 liner shipping services are operated by a vessel fleet of close to 6000 container vessels (WSC2011). The liner shipping industry carries about 60% of goods by value transported internationally by sea (WSC2011). The significance and magnitude of the liner shipping network makes the network design an important transportation problem. The network has high fixed asset costs in terms of the container vessels deployed and hence capacity utilization is crucial to a competitive liner shipping operation. At the same time maritime transport is accountable for an estimated 2.7% of the worlds CO2 emissions, whereof 25% is attributable to container ships alone (WSC2009). Fuel cost is the largest variable cost of operating a liner shipping network (Stopford 1998). Operations research can have a huge impact on the trade of liner shipping as maximising the revenue while considering variable operational cost may ensure a better capacity utilization in the network. Improved capacity utilization will increase profit for liner shipping companies, and give competitive freight rates for global goods. In due time operations research may optimize on reducing the speed of the container fleet to decrease the CO2 emissions from liner shipping in general as seen in the case of tramp shipping (Norstad *et al.* 2009).

The liner shipping network design problem (LSNDP) is to construct a set of non-simple cyclic services to form a capacitated network. The network design maximises the revenue of container transport considering the cost of vessels deployed to services, overall fuel consumption, port call costs and cargo handling costs. Operations research literature on the LSNDP is scarce (Løfstedt *et al.* 2011) compared to related maritime shipping transportation problems, but a surge in publications over recent years show increased interest in the LSNDP. The works of (Agarwal *et al.* 2008, Alvarez 2009, Reinhardt and Pisinger 2011, Plum 2010, Løfstedt *et al.* 2011) reveal that the LSNDP is a very complex optimization problem, where current mathematical formulations and state-of-the-art exact solution methods cannot scale to realistic sized problem instances. One heuristic approach has been applied to large scale instances in (Alvarez 2009, Brouer *et al*. 2011). A core concept in liner shipping is the transhipment of containers. More than 50% of cargoes are transported on more than one service from origin to destination. This means that the LSNDP has an underlying multicommodity flow problem (MCF). (Alvarez 2009) identifies the excessive time used for solving the MCF to evaluate a given network configuration as a bottleneck in local search methods. As a result, within reasonable computation time the tabu search by (Alvarez, 2009) only performs a limited search of the solution space of large scale instances.

In this paper, we present a matheuristic for solving the LSNDP. Matheuristics are an emerging field within optimization and are defined as methods exploiting the synergies of mathematical programming and metaheuristics by (Maniezzo *et al.* 2009). The domain is wide and includes the use of mathematical programming techniques in a heuristic variant as well as deploying mathematical programming methods within a metaheuristic framework (Maniezzo *et al.* 2009). In the present paper we use mathematical programming to explore our neighborhood defined as the solution space of a mixed integer program designed to capture the complex interaction of the cargo allocation between routes. One of the first approaches of using this technique was (Franceschi *et al.*, 2006) for the Distance-Constrained CVRP. The method has also been explored for the Split Delivery VRP by Chen *et al* (2007), the Split Delivery VRP with minimum delivery amounts by Gulczynski *et al. (*2010), by (Archetti *et al.* 2010) for the The Split Delivery Capacitated Team Orienteering Problem and lately in Gulczynski *et al.* (2011) for the Periodic VRP. In all cases the matheuristic solution method combining local search with an integer program as neighborhood has proven very successful compared to other state-of-the-art heuristics.

We make three contributions: We present a construction heuristic based on an interpretation of the LSNDP as a multiple quadratic knapsack problem. Secondly, an improvement heuristic is applied to the solution of the construction heuristic. The improvement heuristic is a large neighborhood search defined by the solution space of a mixed integer program inserting and removing port calls from a single service. Thirdly, the heuristic makes use of estimation functions for the change in a large MCF in order to avoid the bottleneck of solving a large scale MCF. Once moves are applied to a service the neighborhood of subsequent services are based on an optimal solution of the MCF in order to decrease the error of the evaluation functions. The MCF is resolved using an advanced warm start basis and column generation decreasing solution times by up to a factor 40.

The outline of the paper is as follows. In Section 1.1 we review the literature on liner shipping network design. Section 2 describes the individual components of our matheuristic. Section 3 concludes with preliminary computational results and draws perspectives on our projected plans for future work.

1.1 Literature on the LSNDP

Brouer *et al.* (2011) give an introduction to the LSNDP focusing on mathematical modelling of the business domain and the introduction of a benchmark suite of LSNDP problems. Ronen *et. al* (2004) review the field of operations research within shipping in general and a good introduction to the LSNDP may be found in Christiansen *et. Al* (2007). Recently Kjeldsen (2011) published a classification scheme for routing and scheduling problems within liner shipping reviewing and classifying 24 references. The LSNDP was initially studied by Rana and Vickson (1991) as a MIP for a multiple container-ship problem without transshipment and where vessels return to the origin node empty. Benders decomposition principle divides the MIP into an integer network subproblem (INS) and a cargo allocation problem (CAS).

Results are reported for 10-20 ports and three vessels. In recent literature several variants of the LSNDP have been presented. Fagerholt (2004) develops a model and solution method for a regional carrier along the Norwegian coast. The model assumes the carrier loads at a single port and finds optimal routes of vessels to service the unloading facilities. The problem may be dealt with as a VRP problem, given that a designated depot is known and transhipments are not allowed. The solution method is based on complete enumeration solved by a MIP solver. Similarly, Karlaftis *et al.* (2009) solve a problem for the region of the Aegean sea using a genetic algorithm. These models do not deal with transhipments at multiple ports and the resulting interaction between different services.

The simultaneous ship scheduling and cargo routing problem (SSSCR) by Agarwal and Ergun (2008) is based on a time-space network with each port represented on 7 consecutive weekdays. This construction allows non-simple cycles with multiple visits to a port on different weekdays. Computational results are reported for three different heuristics exploiting the separability of solving the route generation problem and the MCF. Results are reported for 6, 10, 15 and 20 ports with up to 100 ships and 114 demands. An important limitation of the SSSCR is that it allows transhipments at no cost.

Reinhardt and Pisinger (2011) presents the LSNDP for a multiple container ship problem with separate routings for each vessel accounting for transhipment costs between routes. The model allows butterfly routes, where two visits are allowed to a single port on a service. A branch-and-cut algorithm is applied to the problem and computational results are reported for 15 ports and up to 6 vessels.

Alvarez (2009) presented the joint routing and fleet deployment model for the LSNDP. The model accounts for transhipment costs and the option of laying up or forward leasing vessels not in use. The model is separable into a service generating problem and a MCF. The overall objective is to maximise the revenue of cargo transported, while considering operational cost of the fleet-, fuel-, transshipment-, and port call-cost. The model is the first to incorporate routings with different speeds in order to optimize on the fuel consumption in the network. Exact solutions are obtained for a six port instance using a MIP solver. Alvarez (2009) describes a tabu search heuristic to solve the problem which is applied to a 120 port instance with a full demand matrix.

Recently, Meng and Wang (2011) presented a mixed integer programming model with the objectives to select among a set of predefined candidate shipping routes, and to select ship deployment to the chosen routes while considering the cargo allocation of full and empty containers regarding the weekly frequency constraint. The model is solved using CPLEX and numerical results are presented for 60 candidate shipping lines, eight vessel types and 600 commodities.

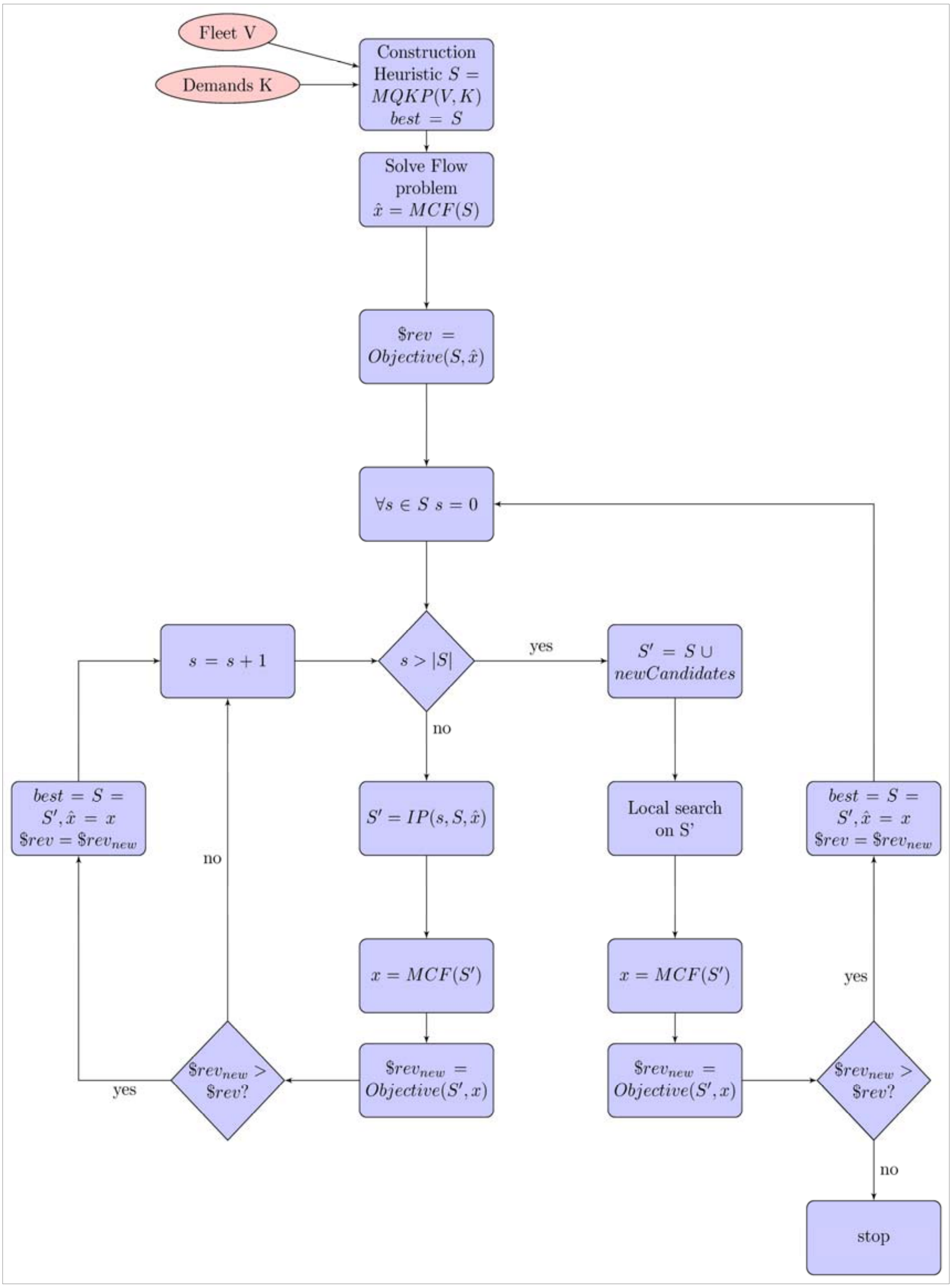


Figure 1. Flowchart of the matheuristic

2 A MATHEURISTIC FOR THE LNSDP

An instance of the LNSDP consists of a set of ports **P,** a set of demands **K**, where each demand has an origin, $\mathbf{O_k} \in \mathbf{P}$, and a destination, $\mathbf{D_k} \in \mathbf{P}$, a set of vessel classes **A** and a number of available vessels $\mathbf{N_a}$ for each class. Each vessel **v** belongs to a given vessel class **a** specifying its capacity $\mathbf{C_a}$, minimum and maximum speed limits, bunker consumption and a weekly sailing distance $\mathbf{W_d}^{\mathbf{a}}$ .The weekly sailing distance is based on the design speed of the vessel, where fuel consumption is optimized. Finally, a distance table of the direct distance $\mathbf{d_{pq}}$ between all pairs of ports $\mathbf{p,q} \in \mathbf{P}$ is given.

A solution to the LSNDP is a set of services **S**. A service is a non-simple cyclic route visiting a set of ports $\mathbf{P'} \subseteq \mathbf{P}$. The rotation time is the time needed to complete the cyclic route including a day for each port call en route for cargo handling. Depending on the vessel class a minimum, $\mathbf{T^a_{MIN}}$, and maximum, $\mathbf{T^a_{max}}$, rotation time in weeks may be defined. It is common in liner shipping to offer a regular service with a weekly frequency. The weekly frequency of port calls is obtained by deploying to a service multiple vessels sailing one week apart. Let $\mathbf{N^a_s}$ be the number of vessels of vessel class $\mathbf{a} \in \mathbf{A}$ deployed to service $\mathbf{s} \in \mathbf{S}$ to maintain a weekly frequency. A service carries a set of demands $\mathbf{k_s} \subseteq \mathbf{K}$ either by serving both $\mathbf{O_k}$ and $\mathbf{D_k}$ or by serving either $\mathbf{O_k}$ or $\mathbf{D_k}$ and a designated transhipment port $\mathbf{G_k}$ valid for transhipping demand $\mathbf{k} \in \mathbf{K}$.

2.1 Algorithmic overview

The matheuristic creates an initial solution using a greedy construction heuristic. The construction heuristic returns a set of services,**S,** that are iteratively improved using an IP for each service to indicate a set of port insertions and removal of each individual service. A local search on the composition of the set of services **S** is wrapped around the loops improving the individual services. An algorithmic overview is illustrated in figure 1.

2.2 Generating an initial solution using a greedy construction heuristic

We obtain an initial solution to the LSNDP by constructing a set of services in which, we place a set of predefined port calls in order to transport the demand. The method is inspired by the multiple quadratic knapsack problem, where a service corresponds to a knapsack and the items are port calls. It is quadratic in the sense that profit is obtained by adding port pairs to the services in order to transport demand. The service set problem is based on a subdivision of the available fleet into a set of services **S** constituted by subsets $\mathbf{S_a} \subseteq \mathbf{S}$ according to vessel classes. It is desirable to have services of varying duration within an interval $[\mathbf{T^a_{MIN}}; \mathbf{T^a_{max}}]$. A random integer $\mathbf{h} \in [\mathbf{T^a_{MIN}}; \mathbf{T^a_{max}}]$ is selected and a service s with $\mathbf{n^a_s=h}$ of **h** weeks duration (an **h**-week rotation requires **h** vessels) is added to the set of services **S**. Set $\mathbf{v=N_a}$. After creation of a new service $\mathbf{s} \in \mathbf{S_a}$ **v** is updated to **v=v-h**. This process is repeated until $\mathbf{v} \leq \mathbf{T^a_{max}}$. If $\mathbf{v} \geq \mathbf{T^a_{min}}$ the final service is created with $\mathbf{h=v} \in [\mathbf{T^a_{MIN}}; \mathbf{T^a_{max}}]$ otherwise we add **h** vessels to the previously created service **s'** possibly exceeding $\mathbf{T^a_{max}}$. If $\mathbf{n^a_{s'}} \geq \mathbf{2...T^a_{MIN}}$ we split the service into two services each with $\mathbf{n^a_s=h/2}$. After this procedure, a set of services **S** is defined, a vessel class **a** and a number of vessels $\mathbf{n^a_s}$ is assigned to each service **s**. The subdivision of the fleet into services means that the initial solution has sunk fixed asset cost by assigning the entire fleet to services. Next we define a set of port calls to place in the services. Each port can be defined as a main port or an outport. The initial solution is limited to the creation of simple cycles. A port call may be placed only once in each service, but in $\mathbf{m_p}$ services. Outports are set with $\mathbf{m_p}$=3, whereas main ports have $\mathbf{m_p}$=10. There is no constraint requiring all port calls placed in the set of services.

The profit of transporting a demand from port $\mathbf{i} \in \mathbf{P}$ to $\mathbf{j} \in \mathbf{P}$ is the revenue $\mathbf{r_k}$ obtained by the transport subtracted the loading and unloading cost $\mathbf{c_l^i}$ and $\mathbf{c_u^i}$ (respectively) of the container en route. A demand transported with no transhipments will have net revenue $\boldsymbol{\rho}_{\mathbf{OkDk}} =(\mathbf{r_k}\text{- } \mathbf{c_l^{Ok}}\text{- } \mathbf{c_u^{Dk}})$ for one unit of **k**. As described in the introduction more than 50% of the demands are transhipped resulting in a MCF. In order to cater for transhipments without considering a MCF the demand matrix is transformed such that each demand is represented by a direct demand and a demand transhipped at a designated transhipment port $\mathbf{G_k}$, where $\boldsymbol{\rho}_{\mathbf{OkGk}}\mathbf{=(1/2r_k}\text{- } \mathbf{c_l^{Ok}}\text{- } \mathbf{c_u^{Gk}})$ and $\boldsymbol{\rho}_{\mathbf{OkDk}}=(\mathbf{1/2r_k}\text{- } \mathbf{c_l^{Gk}}\text{- } \mathbf{c_u^{Dk}})$. This is a simplifying assumption fixing a single transhipment port for each demand to incorporate interaction between services in the construction heuristic. The subsequent improvement heuristic will have no restrictions on transhipment facilities. A port call cost, $\mathbf{c_p^a}$, is associated with a port call depending on the vessel class and a sailing cost is associated with each port pair, $\mathbf{c_{pq}^a}$.

The construction heuristic is a greedy parallel insertion heuristic. The services are seeded with a random number $\mathbf{l} \in \mathbf{\{1;3\}}$ of ports $\mathbf{p} \in \mathbf{P}$. The seeding is either by random or by selecting a port $\mathbf{p} \in \mathbf{P}$ and a transhipment port $\mathbf{q} \in \mathbf{P}$ matched to $\mathbf{p}$. The construction heuristic is based on parallel insertion by a *football teaming principle* i.e. the services take turn at choosing the next port to call. We apply parallel insertion in order to disperse the attractive port call combinations throughout the network. A greedy choice of the most revenue generating port call is made between all feasible port calls with regards to route duration. Feasibility of a given port call is estimated using best insertion in order to respect the weekly frequency constraint, requiring the distance of a route $D(s) \leq W_d^{a(s)} n_s^{a(s)}$, where $\mathbf{a(s)}$ is the vessel class $\mathbf{a} \in \mathbf{A}$ deployed to service $\mathbf{s} \in \mathbf{S}$. The actual routing with regards to distance and capacity utilization is improved using a local search based on simulated annealing and two-opt after assignment of port calls to services by the greedy construction heuristic. The initial solution may have unplaced port calls and excess vessels for services $\mathbf{s}$, where $D(s) \leq W_d^{a(s)}(n_s^{a(s)} - 1)$. Port calls as well as vessels may be included in the solution of the subsequent improvement heuristic. Finally, we apply standard column generation to the MCF of transporting the cargo on the resulting liner shipping network of the initial solution. The solution and dual variables to the MCF is used to calculate the estimation function values of the improvement heuristic.

2.3 Improvement heuristic

Given a solution to the LSNDP $\mathbf{x'}$ with services $\mathbf{S'}$ serving demands $\mathbf{K'} \subseteq \mathbf{K}$ we introduce an integer program to estimate the effect of removing and adding port calls. We define $\mathbf{P^s}$. the set of nodes in the service $\mathbf{s} \in \mathbf{S'}$, $\mathbf{N^s}$ □ $\mathbf{P} \setminus \mathbf{P^s}$; the set of neighbours of a service $\mathbf{s} \in \mathbf{S'}$ defined as nodes within a certain geographical distance of nodes in $\mathbf{P^s}$ and variables:

- $\boldsymbol{\lambda_i}$=1 if item $\mathbf{i} \in \mathbf{P^s}$ is removed from service $\mathbf{s} \in \mathbf{S'}$, 0 otherwise
- $\boldsymbol{\gamma_i}$=1 if item $\mathbf{i} \in \mathbf{N^s}$ □ $\mathbf{P^s}$ is inserted in service $\mathbf{s} \in \mathbf{S'}$, 0 otherwise. If $\mathbf{i} \in \mathbf{P^s}$ the port call represents a reinsertion resulting in a non-simple cycle for the service $\mathbf{s}$.
- $\boldsymbol{\omega_s} \in \mathbf{Z^+}$ an integer variable indicating the number of vessels service $\mathbf{s}$ is expanded with. ω_s can be negative if less vessels are needed after removal of a port call.

We want to make an integer program that removes and inserts port calls in $\mathbf{S'}$, while considering an estimation of the distance travelled on each service (the fleet deployment) and an estimation of the alternative flow of demands arising, when we remove/insert several port calls from/to $\mathbf{S'}$. Routing the cargo is a MCF, but we cannot afford to evaluate the MCF in its entirety and hence we make some simplifying assumptions about rerouting the flow.

When inserting a port call the estimated distance increase is calculated by use of a best insertion heuristic. For each service $\mathbf{s} \in \mathbf{S'}$ we calculate the distance increase $\boldsymbol{\Delta_i^s}$ for each $\mathbf{i} \in \mathbf{N^s}$. Likewise we calculate the decrease of distance $\boldsymbol{\Gamma_i^s}$ for every $\mathbf{i} \in \mathbf{P^s}$. For modelling the distance in-/de-crease of insertions/removals we define the following constants and sets:

- $\boldsymbol{\Delta_i^s}$**:** estimated distance increase for inserting item $\mathbf{i}$ in service $\mathbf{s} \in \mathbf{S}$ according to a best insertion method.
- $\boldsymbol{\Gamma_i^s}$: estimated distance decrease for removing item $\mathbf{i}$ from service $\mathbf{s} \in \mathbf{S}$ joining its predecessor with its successor.
- $\mathbf{E_s}$: set of edges used by the Hamiltonian cycle in service $\mathbf{s} \in \mathbf{S}$.
- $\mathbf{D(s)}$: current distance of the Hamiltonian cycle in service $\mathbf{s} \in \mathbf{S}$.
- $\mathbf{M_a}$**:** number of undeployed vessels of class a in the current service set $\mathbf{S'}$.
- $\mathbf{n_s^a}$: number of deployed vessels of class $\mathbf{a}$ to service $\mathbf{s} \in \mathbf{S'}$.

- $\mathbf{C_v^a}$: cost of deploying a vessel of type $\mathbf{a \in A}$.

Whenever a MIP is solved for some $\mathbf{s \in S'}$ we estimate the effect on the flow in the network. The quality of the flow solution depends on the number of transhipments performed overall in the network and the capacity installed compared to the demand for flow. We define the following estimation functions:

- $\mathbf{\Theta(i)}$: estimated value of inserting a node $\mathbf{i \in N^s}$ in the best insertion position identified when calculating the distance.
- $\mathbf{Y(i)}$: estimated value of removing a node $\mathbf{i \in P^s}$.
- $\mathbf{\Psi(i)}$: estimated value of reinserting a node $\mathbf{i \in P^s}$ by best insertion limited to insertions two port calls away from the current position of **i** in **s**.

In order to estimate the change of the network flow a graph **G=(V,E)** of the residual capacity is constructed, representing the solution **x'** with services **S'** and commodity allocation **K'** mapped onto the network by solving the MCF on **S'**.

Let **|s|** denote the number of unique ports in **s** and let $\mathbf{|P^s|=m}$ denote the number of port calls in a rotation $\mathbf{r^s}$ for **s**, $\mathbf{|P^s|=m \geq |s|}$, $\mathbf{r^s}$ be a rotation defined by the port sequence $p_1^s, p_2^s, \ldots, p_m^s$, **V_p** be the set of port vertices, $\mathbf{V\_r^s}$ be a set of vertices representing the port call sequence for rotation $\mathbf{r^s}$. **V_r=** V_ $\mathbf{r^s}$ is the set of rotation vertices representing all port calls by all rotations and the set of vertices is defined as **V=V_p** **V_r** be the set of vertices.

Let the set of edges $E = E_l \cup E_d \cup E_v$ be the set of edges, where

- E_l $\mathbf{=\{(p,v) | p \in V\_p, v \in V\_r^s\}}$ is the set of load edges representing a departure from port **p** to the rotation $\mathbf{r^s}$.
- E_d $\mathbf{=\{(v,p) | v \in V\_r^s, p \in V\_p\}}$ is the set of discharge edges representing an arrival at port **p** from the rotation $\mathbf{r^s}$.
- E_v $\mathbf{=\{(v,u) | v,u \in V\_r^s, v=p^s_h, u==p^s_{(h+1) \bmod m}\}}$ is the set of voyage edges representing a voyage between two consecutive port calls in $\mathbf{r^s}$.

Let $\mathbf{C_e}$ be the capacity of edge $\mathbf{e \in E}$, where $\mathbf{C_e =}$ ∞ for $\mathbf{e \in}$ $E_l \cup E_d$ and $\mathbf{C_e}$, e∈ E_v be the residual capacity of edge **e** after flow assignment of the MCF onto **S'**.

- $\mathbf{c_e}$ be the edge cost, where $\mathbf{c_e=0, e \in}$ E_v (as the cost is on the vessel) and let $\mathbf{c_e = c_l^P, e}$ $\in$ E_l and $\mathbf{c_e = c_u^P, e \in}$ E_d be the cargo handling cost of loading or unloading a container at port $\mathbf{p \in V\_P}$, where **p** is either the source or the target of the edge respectively.

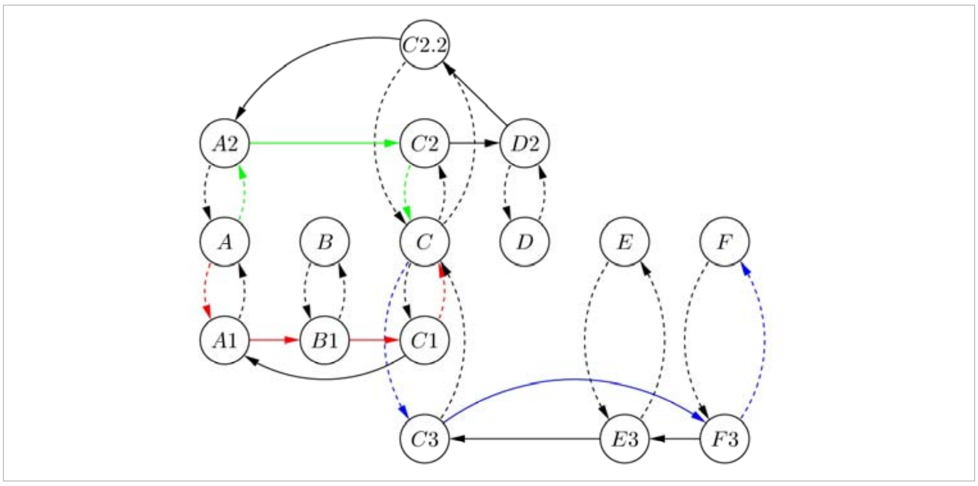


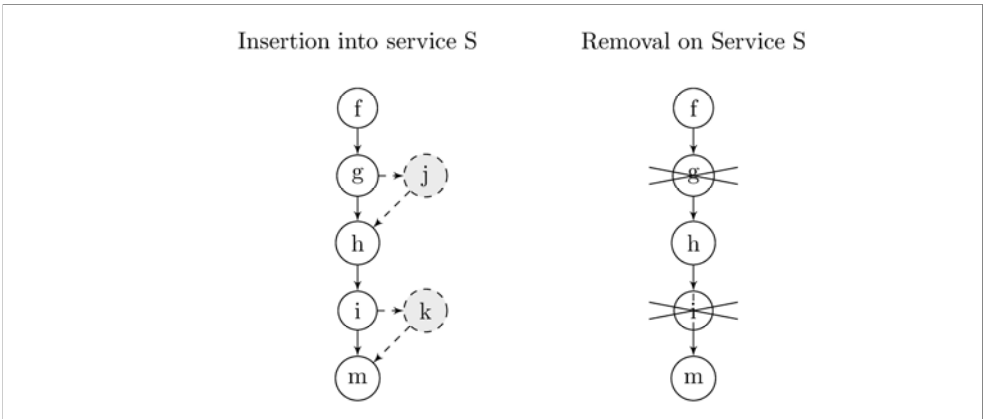
Figure 2: An example of a hub and spoke network with 1 hub, C, and 5 spokes (A,B,D,E,F) and 3 rotations

The estimated value of insertion - **Θ(i):**When we insert a port call vertex $\mathbf{i^s} \in \mathbf{N^s}$ with corresponding port vertex i ∈ V_p in the position between nodes h^s and l^s the demands of the set $\mathbf{K_i}$**={ k ∈ K| i=** $\mathbf{O_k}$ **v** $\mathbf{D_k}$**}** becomes eligible for transport using service **s ∈ S'**. Solving a shortest path problem on **G'** where **V'_** $\mathbf{r^s}$ $\cup$ $\{\mathbf{i^s}\}$, and $E'_v = E_v \setminus \{(h^s l^s)\} \cup \{(h^s i^s),(i^s l^s)\}$, $E'_l = E_l \cup \{(i i^s)\}$, $E'_d = E_d \cup \{(i^s i)\}$ will identify for each $\mathbf{k} \in \mathbf{K_i}$ whether there is an (improved) path for **k** in **G'** in terms of transhipment costs (TC), the increase of revenue in demand transported (RK) and the capacity available. If the capacity is not available the largest bottleneck can be identified and the largest or least profitable demand on this bottleneck may be rerouted in **G'** where the path is removed. The estimated value **Θ(i)** should account for in-/de-crease in transhipment cost, in-/de-creased revenue of the flow, and increase in port call cost: **Θ(i)= TC(G',** $\mathbf{K_i}$**)-TC(G,** $\mathbf{K_i}$**) + RK(G')-RK(G)-** $\mathbf{c_i^s}$.

The estimated value of removal - **Y(i)**: When a port call vertex $\mathbf{i^s} \in \mathbf{P^s}$ is removed between nodes h^s and l^s, commodities of the set K_i transported on **s** must be rerouted or omitted. Define $\mathbf{K_i^s}$**={ k ∈** $\mathbf{K_i}$**| k is transported on s}**. **Y(i)** estimates rerouting $\mathbf{K_i^s}$ in the remaining network by solving a shortest path problem on **G'=(V_** $\mathbf{r^s \setminus \{i^s\}}$**,** $E_v \setminus \{(h^s i^s),(i^s l^s)\} \cup \{(h^s l^s)\}$**)**. G' will identify for each k ∈ $\mathbf{K_i^s}$ whether there is an alternative path in the network. The estimated value **Y(i)** should account for the in-/de-crease in transhipment cost TC for each commodity k ∈ $\mathbf{K_i^s}$ rerouted in G', and the decrease of revenue flow RK for omitted cargo and the decrease in port call cost. **Y(i)=TC(G',** $\mathbf{K_i}$**)-TC(G,** $\mathbf{K_i}$**) + RK(G')-RK(G) -** $\mathbf{c_i^s}$.

The estimated value of reinsertion - **Ψ(i)**:When a node $\mathbf{i} \in \mathbf{P^s}$ is reinserted the set $\mathbf{K_i^s}$ may have alternative shorter paths on **s**. As a result the residual capacity of one or more edges in E_s will increase. **Ψ(i)** estimates whether the edges with increased capacity will result in an improved solution to the MCF. Let $E_{K_i} \subseteq E_s \subseteq E_v$ be the set of edges with increased capacity and δ_e be the capacity increase. The dual value $\pi_e \in E_{K_i}$ in the solution of the MCF indicates the increase in revenue for each unit of additional capacity. The estimated value **Ψ(i)** accounts for the expected revenue increase and the increase in port call cost for reinserting **i**. **Ψ(i)=** $\sum_{e \in E_{K_i}} \delta_e \pi_e + c_i^s$.

Lock sets: The MIP is solved for a single service with the remaining services fixed. A solution to the MIP may result in several insertions and removals referred to as a *move* in the following. The estimation functions are based on performing a particular move without consideration for additional removals/insertions. In order to reduce the error of the estimation functions we define *lock sets* of a move constraining insertions/removals on port calls related to a move. When inserting a port call **i**, a set of new commodities $\mathbf{K_i}$ may be transported. The origins and destinations of $\mathbf{k} \in \mathbf{K_i}$ should not be removed. The estimation function relies on the residual capacity of the remaining network. Insertions before bottlenecks introduced by the routing of $\mathbf{K_i}$ should be avoided. We define the set of *Insertion locks* on inserting $\mathbf{i} \in \mathbf{N^s}$ as $\mathbf{L(i^+)}$. $\mathbf{L(i^+)}$ places a lock on removal of origin/destination nodes ($\mathbf{i} \in \mathbf{P^s}$) for $\mathbf{k} \in \mathbf{K_i}$, and lock on insertion of nodes ($\mathbf{i} \in \mathbf{N^s}$) with best insertion position before bottlenecks introduced by routing $\mathbf{K_i}$.



(a) Insertion of port call vertex. The dashed nodes j and k are considered for insertion

(b) Removal of a port call vertex. The crossed out nodes g and l are considered for removal

Figure 3: Insertion and removal moves for the matheuristic

When we reinsert a port call we introduce a non-simple cycle. The estimated value of reinserting a port call is the reduced cost of increasing capacity between the port calls to **i**. We introduce *Reinsertion Locks* for reinserting $\mathbf{i} \in \mathbf{P^s}$ as $\mathbf{L(i^2)}$ locking removal of port calls between two identical port calls to **i** as well as the original port call to **i**. The total number of removals from a service is constrained to $\mathbf{F_s}$. $\mathbf{F_s}$ is dependent on the number of port calls on a service at a low value to reduce the error of the distance decrease function Γ_i^s.

2.4 MIP formulation

The following MIP optimizes a single service and suggests a set of removals and insertions of port calls. The function **a(s)** returns the vessel class assigned to service **s**.

$$max \sum_{i \in N^s} \Theta(i)\gamma_i + \sum_{i \in P^s} (\Upsilon(i)\lambda_i + \Psi(i)\gamma_i) - C_v^{a(s)}\omega_s \tag{1}$$

$$\text{subject to:} \quad D(s) + \sum_{i \in N^s} \Delta_i^s \gamma_i - \sum_{i \in P^s} \Gamma_i^s \lambda_i \le W_d^{a(s)}(n_s^a + \omega_s) \tag{2}$$

$$\omega_s \le M_{a(s)} \tag{3}$$

$$\sum_{i \in P^s} \lambda_i \le F_s \tag{4}$$

$$\sum_{j \in L(i^+)} \gamma_j + \lambda_j - 2|L(i^+)|(1-\gamma_i) \le 0 \qquad \forall i \in N^s \tag{5}$$

$$\sum_{j \in L(i^2)} \lambda_j - |L(i^2)|(1-\gamma_i) \le 0 \qquad \forall i \in P^s \tag{6}$$

$$\lambda_i \in \{0,1\} \qquad \forall i \in P^s \tag{7}$$

$$\gamma_i \in \{0,1\} \qquad \forall i \in N^s \cup P^s \tag{8}$$

$$\omega_s \in \mathcal{Z} \tag{9}$$

The objective function (1) maximises the benefit obtained from removing and inserting several port calls accounting for the estimated change of revenue, transhipment cost, port call cost and fleet cost. The number of vessels needed to maintain weekly frequency on the service after insertion/removal is estimated in Constraint (2). Constraint (3) ensures that the solution does not exceed the available fleet of vessels. Constraint (4) ensures that we can only remove $\mathbf{F_s}$ nodes from the service. The set of nodes that are affected by the insertion move are fixed by Constraints (5). Constraints (6) ensure that we cannot remove nodes between two identical port calls when we reinsert a port **i** already in $\mathbf{P^s}$.

Algorithm 1 MatHeuristic(I)

Require: An Instance I of the LSNDP (S, P, A, D, K)

1: $S' \leftarrow GreedyLSNDP(I)$
2: $x' \leftarrow MCF(S', K)$
3: $IMPROVE \leftarrow true$
4: **while** IMPROVE **do**
5: $\quad IMPROVE \leftarrow false$
6: $\quad$ **for all** $s \in S'$ **do**
7: $\quad\quad$ **for all** $i \in N^s$ **do**
8: $\quad\quad\quad L(i^+), \Theta(i) \leftarrow CalcInsert(i, S')$
9: $\quad\quad$ **end for**
10: $\quad\quad$ **for all** $i \in P^s$ **do**
11: $\quad\quad\quad \Upsilon(i) \leftarrow CalcRemoval(i, S')$
12: $\quad\quad\quad L(i^2), \Psi(i) \leftarrow CalcReInsertion(i, S')$
13: $\quad\quad$ **end for**
14: $\quad\quad IP \leftarrow constructMIP(s, P^s, N^s, \Theta, \Upsilon, \Psi, \bigcup_{i \in N^s \cup P^s} L(i^+), \bigcup_{i \in P^s} L(i^2))$
15: $\quad\quad M \leftarrow solve(IP)$
16: $\quad\quad$ **if** M != $\emptyset$ **then**
17: $\quad\quad\quad \bar{S} \leftarrow ApplyMoves(M, L(i^s_-), s)$
18: $\quad\quad\quad \bar{x} \leftarrow \Delta MCF(S', K)$
19: $\quad\quad$ **end if**
20: $\quad\quad$ **if** $obj(\bar{x}) \geq obj(x')$ **then**
21: $\quad\quad\quad x' \leftarrow \bar{x}$
22: $\quad\quad\quad S' \leftarrow \bar{S}$
23: $\quad\quad\quad IMPROVE \leftarrow true$
24: $\quad\quad$ **end if**
25: $\quad$ **end for**
26: **end while**

Algorithm 1 gives an overview of the matheuristic. The initial solution is constructed by the greedy parallel insertion GreedyLSNDP(I) of an instance I in line 1. The resulting MCF is solved in line 2. The improvement heuristic loops over the set of services **S**. The estimation functions and lock sets for **s** are calculated in lines 7-13. The MIP (1)-(9) for **s** is constructed and solved in lines 14-15. The solution is evaluated by resolving the new MCF in line 18. Δ MCF(S',K) is a column generation algorithm for the MCF using a warm start basis. The basis consists of all commodities and services not directly affected by the moves identified by the MIP. The algorithm ΔMCF(S',K) has been experimentally evaluated to decrease solution times by a factor 5-40 depending on the number of commodities affected by the move and also the number of moves applied. If the solution is improved the new solution is saved in lines 20-23 before the next MIP is calculated for the following **s** ∈ **S'**. The algorithm terminates when an entire loop over the set of services **S** does not result in an improved solution. The loop structure over the set of services **S'** above depends on all preliminary moves made. A simulated annealing procedure allows non-improving moves to diversify the search. Algorithm1 is meant to be embedded in a simple local search scheme to adjust the number of services in the solution as seen in the flowchart of figure 1 on page 4.

3 CONCLUSION

Preliminary computational results for a version without the local search loop have been performed using the benchmark suite from Brouer *et al.* (2011). The preliminary computational results indicate that the method scales well. The IP models are small with less than 50 binary variables and few constraints. The size of the IPS do not increase significantly as the services only increase slightly in size for large instances and the IPS are solved by CPLEX in less than a second. Resolving the multicommodity flow takes 10-20 seconds for large instances. However, the solution quality without the local search on the composition of services leads to low capacity utilization for large parts of the network and some demands are not transported because the vessels deployed here can not be efficiently reallocated by deleting unpromising services. This causes the search to get trapped in a local minimum and to converge before finding a good solution. The algorithm is still promising as it is possible to search among many different solutions but it is necessary to implement further local search methods in order to get a good composition of services to cover the demands. Future work will concentrate on this local search method to improve upon results.

ACKNOWLEDGEMENTS

This project was supported by The Danish Strategic Research Council under the ENERPLAN project.

REFERENCES

Agarwal, R., and Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, **42(2**), 175-196.

Alvarez J. (2009) Joint routing and deployment of a fleet of container vessels. *Maritime Economics and Logistics* , **11(2)**:186.

Archetti C., Bianchessi, N., Hertz, A. and Speranza M.G. The split delivery capacitated team orienteering problem. *Technical Report 55, GERAD, 2010. Les Cahiers du GERAD.*

Brouer, B., Alvarez J., Plum, C., Pisinger, D and Sigurd, M.M (2011). An integer programming model and benchmark suite for liner shipping network design. Technical report, DTU

Chen, S., Golden, B., and Wasil, E. (2007). The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, **49(4)**, 318-329.

Christiansen, M., Ronen, D., Nygreen, B., and Fagerholt, K. (2007). Maritime transportation. *Handbooks in Operations Research and Management Sciences*, **14**, 189-284.

Fagerholt, K. (2004}). Designing optimal routes in a liner shipping problem. *Maritime Policy & Management*, **31(4)**, 259.

Franceschi, R. D., Fischetti, M., and Toth, P. (2006). A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming*, **105(2)**, 471-499.

Gulczynski, D., Golden, B., and Wasil, E. (2010). The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review*, **46(5)**, 612-626.
Gulczynski, D., Golden, B., and Wasil, E. (2011). The period vehicle routing problem: New heuristics and real-world variants. *Transportation Research Part E: Logistics and Transportation Review*, **47(5)**, 648-668.
Karlaftis, M. G., Kepaptsoglou, K., and Sambracos, E. (2009). Containership routing with time deadlines and simultaneous deliveries and pick-ups. *Transportation Research Part E*, **45**, 210-221.
Kjeldsen, K. H. (2011). Classification of ship routing and scheduling problems in liner shipping. *Infor*, **49(2)**, 139-152.
Maniezzo, V. and Stützle, T. and Voβ, S. (2010) Matheuristics: Hybridizing metaheuristics and mathematical programming, *Annals of Information Systems*, **10**
Meng, Q., and Wang, S. (2011). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, **47(5)**, 695-708.
Norstad, I., Fagerholt, K., and Laporte, G. (2011) Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, **19(5),** 853–865.
Plum C. (2010) The linearized simultaneous string-design and cargo-routing problem. *EURO 2010 conference proceedings*
Rana, K., and Vickson, R. G. (1991). Routing container ships using lagrangian relaxation and decomposition. *Transportation Science*, **25(3)**, 201-214.
Reinhardt, L. B., and Pisinger, D. (2011). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, in press, 1-26.
Ronen, D., Fagerholt, K., and Christiansen, M. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, **38(1)**, 1-18.
Stopford, M. (1998). *Maritime Economics*,second editon, Palgrave Macmillan.
Wang, S., and Meng, Q. (2012). Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E: Logistics and Transportation Review*, **48(2)**, 470-484.
Wang, S., and Meng, Q. (2012). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, **48(3)**, 701-714.
The World Shipping Council (WSC) (2009) The liner shipping industry and carbon emissions policy. www.shippingandco2.org/LinerShippingandCO2EmissionsPolicySeptember.pdf, Last viewing September 2011.
The World Shipping Council (WSC) (2011) How liner shipping works. *http://www.worldshipping.org/about-the-industry/how-liner-shipping-works*, Last viewing September 2011.

BENEFITS OF A TRUCK APPOINTMENT SYSTEM ON THE SERVICE QUALITY OF INLAND TRANSPORT MODES AT A MULTIMODAL CONTAINER TERMINAL

Elisabeth Zehendner[1] and Dominique Feillet[1]

[1]*Ecole des Mines de Saint-Etienne, CMP Georges Charpak, F-13541 Gardanne, France*

ABSTRACT

The connection of a container terminal to its hinterland is a key area for competition. Therefore, more and more attention is paid to the service quality of inland transport modes such as trucks, trains and barges. An efficient allocation of internal material handling resources and the use of new strategies, such as truck appointment systems, can reduce the time vehicles spend at the terminal. We propose a mixed integer linear programming model, based on a network flow representation of the terminal, to determine the number of appointments to accept per time slot and an allocation of internal resources minimizing service times of trains and barges simultaneously. By comparing container terminals with and without appointment systems, we show that a truck appointment system is beneficial for trucks as well as for trains and barges.

Keywords: container terminal, intermodal transportation, resource allocation, truck appointment system

1 INTRODUCTION

In view of global supply chains, container terminals are increasingly competing not as individual places but as crucial links within supply chains. The connection of a terminal to its hinterland becomes a key area for competition. Delays at the terminal should be minimized since they downgrade the overall productivity of the freight transportation system. Increasing the throughput of a terminal by extending its transfer and storage facilities is expensive and limited by space restrictions. Consequently, container terminals have to use their existing resources more efficiently and use new strategies, such as truck appointment systems, to reduce transfer times.

Truck appointment systems are used to reduce congestion at the terminal by limiting the number of trucks admitted per time slot and evening out the demand over the day. Maguire *et al* (2010), Morais and Lord (2006), Srour *et al* (2003) and Giuliano and O'Brien (2007) show via case studies that appointment systems have the potential to improve operations inside the terminal dramatically. Terminals get a better visibility on the moves per day. The trucking community will benefit from a higher productivity resulting from faster turnaround times at terminals. But, these studies also highlight the fact that the success of an appointment system depends on the way it is implemented and relies on a large percentage of trucks using it.

Other studies propose methodologies to determine the number of trucks to accept per time slot. Huynh and Walton (2008) propose a combination of mathematical formulation and simulation based on resource constraints and target truck turnaround times. Murty *et al* (2005) mention a simulation model minimizing a combined penalty for yard-crane idle time and for the fraction of time during which the queue of trucks waiting at the block for service from the yard crane is too long. Chen *et al* (2011) develop a convex nonlinear programming model minimizing the total truck turnaround time and the difference between preferred arrival times and assigned time slots. In a second step, they determine time-varying tolls that lead to the optimized truck arrival pattern. Guan and Liu (2009) use a multi-server queuing model and a nonlinear optimization model to determine the optimal number of gate lanes to open while minimizing a combined cost of truck waiting times and gate operating costs.

These studies represent only trucks with their loading and unloading activities and do not include other transport modes like trains, barges and vessels. They assume that the capacity available to serve trucks per time slot is given. Our study is based on container terminals sharing internal transport resources (e.g., straddle carriers) among trucks, trains, barges and vessels. In this case, the number of proposed appointments and the number of resources allocated to trucks are related. Our objective is to determine a resource allocation proposing appointments close to the preferred arrival pattern of trucks while minimizing the delays of trains, barges and vessels. This enables us to see the impacts of a truck appointment system on the service quality of trucks, trains and barges.

Section 2 introduces the analyzed problem and the kind of terminals to which our study applies. Section 3 presents the general modeling approach, as well as an exemplary implementation of a terminal at the Grand Port Maritime de Marseilles. Section 4 evaluates the impact of a truck appointment system on the service quality of truck as well as of trains, barges and vessels. Delays at a terminal with a truck appointment system are compared to delays at a terminal without an appointment system. This comparison is done with results obtained by the optimization model and by discrete-event simulation.

2 PROBLEM DESCRIPTION

Container terminals play the role of exchange hubs in intermodal transportation. They offer transfer facilities to move containers from vessels to trucks, trains and barges and vice versa. A terminal is composed of quays, the inland area and the yard. Figure 1 illustrates an exemplary container terminal. Vessels berth at the quay where containers are unloaded and loaded by quay cranes. The inland area is the terminal's interface with the inland transportation system (rail, road and waterway). It provides truck and train receiving gates where rail cars and trucks are unloaded and loaded with the appropriate equipment. Barges may be served at specific barge gates or at the same quays than vessels. The yard serves as a temporary storage location for full and empty containers. The storage and transportation operations between the different areas are executed by specific equipment.

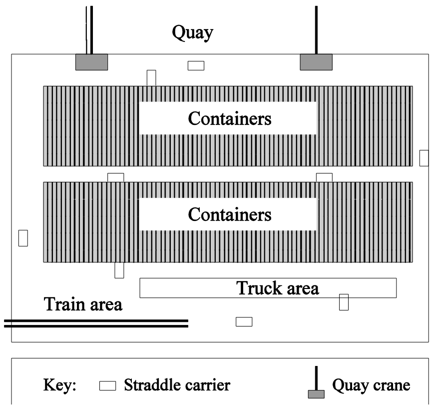


Figure 1. Schematic view of a container terminal

We concentrate our study on container terminals using only manned straddle carriers for internal transportation of containers and for storage operations at the yard. We assume that their drivers work during fixed shifts and may be hired the day before. For organizational reasons, manned straddle carriers are assigned to one transport mode or vehicle (e.g., serve trucks or serve a vessel) for a given time interval. This assignment can only be altered after a given time period.

Trucks have to book an appointment for a specific time slot to enter the terminal to deliver or pickup a container. Entering the terminal without an appointment is not possible. In return and to minimize truck waiting times, each truck is served within a guaranteed service time. To minimize truck drivers discomfort the proposed appointments should match the preferred arrival pattern of trucks.

Trains, barges and vessels differ in cargo volume, operating costs, knowledge and reliability of arrival dates, due dates and required handling equipment. They are therefore served at specific gates, with dedicated equipment and an adapted service strategy. These strategies have to be included in the optimization model to accurately represent the terminal. Possible service strategies are for example serve as fast as possible, serve within a given time window or penalize non-executed tasks at the vehicle's departure.

Our objective is to determine the number of appointments to offer for trucks and a resource allocation of straddle carriers to trucks, trains, barges and vessels. The proposed solution should minimize truck drivers discomfort and delays of trains, barges and vessels. The resulting delays will be compared to the delays at a terminal without appointment system.

2 MIXED INTEGER LINEAR PROGRAMMING MODEL FOR THE RESOURCE ALLOCATION PROBLEM

Studies addressing the resource allocation problem at container terminals without truck appointment systems use either network flow models (e.g., Gambardella *et al,* 2001; Kozan, 2000; Zehendner *et al,* 2011), queuing models (e.g., Kang *et al*, 2008; Alessandri *et al,* 2008) or the vertex-disjoint path cover problem (e.g., Vis *et al*, 2005). In Zehendner *et al* (2011), we chose, similar to Gambardella *et al* (2001), to implement our model as a network flow model where containers to be moved are flows in the network and arc capacities are limited by the number of allocated resources. In this paper, we present an adaptation of this model to include a truck appointment system.

In this section, we present our mixed integer linear programming model used to determine a resource allocation minimizing deviations between preferred truck arrivals and assigned appointments as well as delays of trains, barges and vessels. The underlying idea of the modeling approach is presented first. Then, we provide a formal representation of an exemplary container terminal. Our approach is general enough to adapt the model easily to other terminals operating with straddle carriers.

2.1 Modeling approach

We model the container terminal as a network flow where containers to be moved are flows in the network and arc capacities are limited by the number of allocated straddle carriers. Figure 2 illustrates the network flow model used to determine an assignment of trucks to time slots and the number of straddle carriers to be allocated to trucks at each period. Figure 3 presents the network flow model used to determine a resource allocation to different vehicles minimizing their delays.

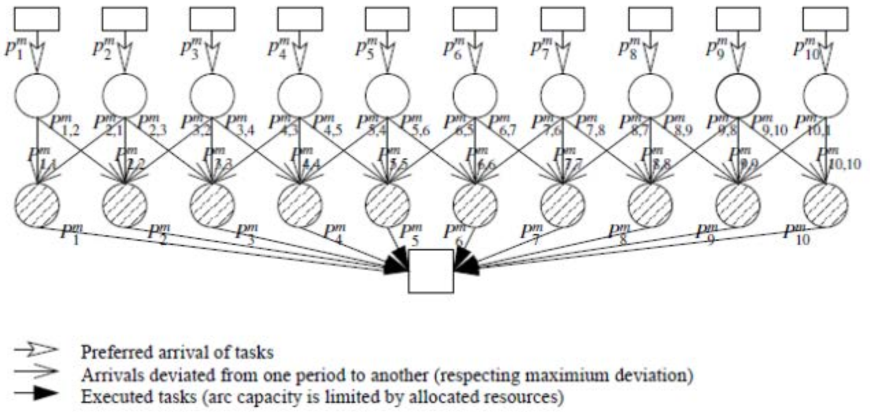


Figure 2. Truck appointment system

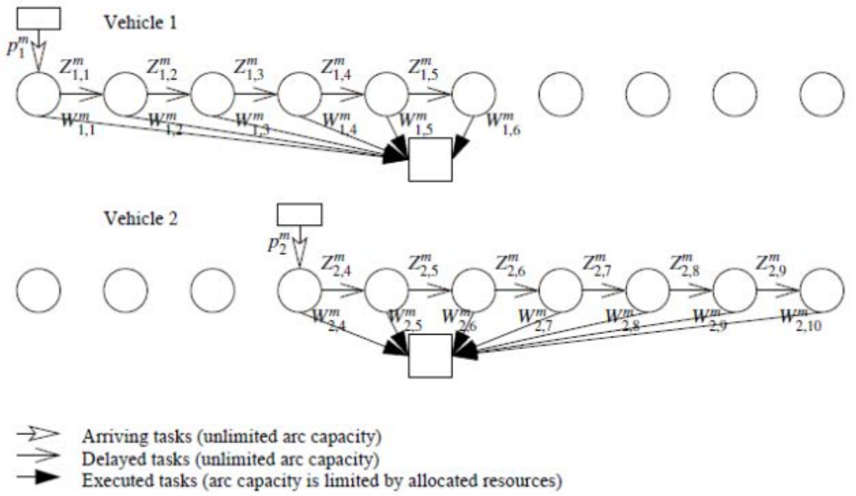


Figure 3. Allocation of straddle carriers to vehicles

In Figure 2, the circle nodes represent the discrete time periods of the working day. Flows p^m_r represent the number of trucks that wish to enter the terminal per period to pick up or deliver a container. Flows $P^m_{r,t}$ assign each truck wishing to enter the terminal to a time slot during which it should be served. Here, the maximum deviation between a truck's preferred arrival and its assigned time slot is limited to plus minus one period. Flows P^m_t represent the total number of trucks to be served per period. These flows indicate the number of appointments to offer per period and are limited by the capacity of straddle carriers allocated to trucks at each period.

Figure 3 represents two vehicles (e.g., trains, barges or vessels) arriving at the terminal. Flows p^m_i represent the arrival of the vehicles at periods 1 and 4 with their demand for container movements. Flows $W^m_{i,t}$ represent the number of executed container movements for each vehicle per period. These flows are limited by the capacity of allocated straddle carriers. Flows $Z^m_{i,t}$ represent the number of tasks per vehicle not executed at the end of a period. These tasks are transferred to be served in the next period. Vehicles should be served prior to their departure and no tasks are delayed after periods 6 and 10.

To represent the entire terminal the truck submodel and the network flow models for all vehicles are related by one constraint limiting the number of allocated straddle carriers. Only this constraint relates the otherwise independent submodels. The modular structure of the problem enables us to design independent submodels for each vehicle or transport mode. Additional parameters, constraints and variables are added to each submodel in order to represent the specific characteristics of different vehicles and transport modes. This modularity makes it easy to adapt the model to different container terminals.

2.2 Exemplary implementation

We present an exemplary implementation for one of the container terminals at the Grand Port Maritime de Marseilles. This container terminal serves trains, barges, vessels and trucks with dedicated service strategies. Rail cars stay at the terminal over the day and are picked up by an engine according to a fixed schedule every day. There is a cost for each container remaining at the terminal after the departure of its train. Barges should be served as fast as possible after their arrivals. Vessels have to be served during their previously defined time windows with a non-increasing number of straddle carriers. Currently the terminal is operating without a truck appointment system. Like Chen *et al.* (2011), we assume that trucks may only enter the terminal with an appointment and aim to propose appointments close to the preferred arrival pattern of trucks.

The input parameters of the model are the expected workload and the capacity of the terminal. The expected workload is determined by the number of vehicles with their arrival times, their due dates and their number of required container movement requests. The terminal's capacity is given by the number of available straddle carriers and the average number of containers a straddle carrier can handle per period. This data is described by the following parameters:

General parameters:

T	Number of time periods describing the time horizon (e.g., one working day)
t	Index of a time period, $t = 1,\ldots,T$
$\mathtt{T}$	Set of all time periods, $\mathtt{T} = \{1,\ldots,T\}$
m	Index of a transport mode: $m = v$ for vessels, $m = b$ for barges, $m = r$ for trains and $m = c$ for trucks
$\mathtt{M}$	Set of all transport modes, $\mathtt{M} = \{v, b, r, c\}$
k_t	Number of available straddle carriers per period t
h^m	Average number of tasks a straddle carrier serving transport mode m can handle per period ($h^m \geq 1$ and $h^m \in \mathrm{N}^+$ for all $m \in \mathtt{M}$)

Parameters for trucks:

r	Index of a time period related to trucks preferred arrivals, $r = 1,\ldots,T$
t	Index of a time period related to trucks assigned arrivals, $t = 1,\ldots,T$
$p^m{}_r$	Aggregated number of trucks wishing to arrive in period r
Δ	Maximal allowed deviation (+/-) from preferred arrival time
$w^m{}_{r,t}$	Cost for deviating the arrival of a truck from period r to period t, $w^m{}_{r,t} = 0$ if $t = r$ and $w^m{}_{r,t} > 0$ otherwise

Parameters for trains, barges and vessels:

I^m	Number of vehicles of transport mode m arriving during the time horizon
i	Index of a vehicle of transport mode m, $i = 1,\ldots,I^m$
$\mathtt{I}^m$	Set of all vehicles of transport mode m, $\mathtt{I}^m = \{1,\ldots,I^m\}$
$r^m{}_i$	Period t in which vehicle i of transport mode m arrives at the terminal
$d^m{}_i$	Period t in which vehicle i of transport mode m has to be ready for departure ($d^m{}_i \leq T$)
$p^m{}_i$	Total number of tasks to be carried out for vehicle i of transport mode m
$q^m{}_i$	Maximum number of containers that may be loaded/unloaded per period per vehicle i of transport mode m
w^m	Delay costs of transport mode m

The model's output is an allocation of straddle carriers to transport modes and vehicles, the number of container movements to be executed per vehicle per period and the number of trucks to be accepted per period. The delays and deviations resulting from the proposed allocation are also determined. This information is given by the variables below. If not indicated otherwise all variables are real.

Variables for trucks:

X^m_t Number of straddle carriers allocated to trucks in period t

$P^m_{r,t}$ Number of trucks wishing to arrive in period r and assigned to period t

P^m_t Total number of trucks assigned to period t (limited by the capacity of allocated straddle carriers)

Variables for trains, barges and vessels:

$X^m_{i,t}$ Number of straddle carriers allocated to vehicle i of transport mode m in period t (real or integer variable)

$W^m_{i,t}$ Number of tasks executed for vehicle i of transport mode m in period t (limited by the capacity of allocated straddle carriers)

$Z^m_{i,t}$ Number of non-executed tasks for vehicle i of transport mode m in period t which are transferred to period $t + 1$

$S^m_{i,t}$ Binary variable indicating if the service of a vehicle has already started prior to period t (t excluded)

$Y^m_{i,t}$ Binary variable indicating if the vehicle is completely served (unloaded and loaded) at the end of period t

U^m_i Number of container movement requests that are not executed at the vehicle's departure

The entire container terminal is represented by the mixed integer linear programming model below. The objective function minimizes delays of barges and trains at the terminal as well as the deviations of trucks. Delays are measured differently for the different transport modes. Planning the delay of vessels is not allowed. Therefore, vessels do not appear in the objective function. Barges may not leave the terminal until they are completely served. Each period they spend at the terminal is penalized. Trains leave at a prefixed time and all non-executed tasks are penalized. Deviations between preferred truck arrivals and assigned time slots are also penalized (remember that $w^c_{r,t} = 0$ if $r = t$). Weights are added to take into account the different volumes of vehicles from different transport modes and possible priorities among them.

$$\min w^b \cdot \sum_{i=1}^{I^b} \sum_{t=r_i^b}^{d_i^b} Y^b_{i,t} + w^r \cdot \sum_{i=1}^{I^r} U^r_i + \sum_{r=1}^{T} \sum_{t=1}^{T} w^c_{r,t} \cdot P^c_{r,t}$$

s.t.

$$p_r^m = \begin{cases} \sum_{t=1}^{r+\Delta} P_{r,t}^m & \forall r = 1, \ldots, \Delta, \\ \sum_{t=r-\Delta}^{r+\Delta} P_{r,t}^m & \forall r = \Delta + 1, \ldots, T - \Delta, \qquad m = c \\ \sum_{t=r-\Delta}^{T} P_{r,t}^m & \forall r = T - \Delta + 1, \ldots, T \end{cases} \tag{1}$$

$$P_t^m = \begin{cases} \sum_{r=1}^{t+\Delta} P_{r,t}^m & \forall t = 1, \ldots, \Delta, \\ \sum_{r=t-\Delta}^{t+\Delta} P_{r,t}^m & \forall t = \Delta + 1, \ldots, T - \Delta, \qquad m = c \\ \sum_{r=t-\Delta}^{T} P_{r,t}^m & \forall t = T - \Delta + 1, \ldots, T \end{cases} \tag{2}$$

Constraint (1) makes sure that each truck is assigned to arrive during exactly one time slot within its maximum deviation range. Constraint (2) determines the total number of trucks assigned to each time period. Both constraints include the fact that no arrivals and assignments take place before or after the time horizon.

$$P_t^m \leq h^m \cdot X_t^m \qquad m = c, \forall t \in \mathcal{T} \tag{3}$$

$$W_{i,t}^m \leq h^m \cdot X_{i,t}^m \qquad m = v, b, r, \forall i \in \mathcal{I}^m, t \in \mathcal{T} \tag{4}$$

$$P_t^m \geq h^m \cdot (X_t^m - 1) + 1 \qquad m = c, \forall t \in \mathcal{T} \tag{5}$$

$$W_{i,t}^m \geq h^m \cdot (X_{i,t}^m - 1) + 1 \qquad m = v, b, r, \forall i \in \mathcal{I}^m, t \in \mathcal{T} \tag{6}$$

Constraints (3) and (4) limit the number of executed tasks per transport mode or vehicle per period by the capacity of straddle carriers allocated to this transport mode or vehicle. Constraints (5) and (6) impose that each allocated straddle carrier executes at least one task. They make the solution more comprehensible by preventing the allocation of excess resources.

$$W_{i,t}^m \leq q^m \qquad m = v, b, \forall i \in \mathcal{I}^m, t \in \mathcal{T} \tag{7}$$

Vessels and barges have to be loaded/unloaded by quay cranes. Straddle carriers can lift up containers on their own from the ground. Therefore, they may operate independently from the quay cranes. Straddle carriers and quay cranes are only linked by the transported volume. Constraint (7) makes sure that the number of transported containers per period does not exceed the number of containers that quay cranes can handle per period.

$$Z_{i,t}^m = \begin{cases} p_i^m - W_{i,t}^m & t = r_i^m, \\ Z_{i,t-1}^m - W_{i,t}^m & \forall t = r_i^m + 1, \ldots, d_i^m \end{cases} \qquad m = v, b, r, \tag{8}$$

$$Z_{i,d_i^m}^m = 0 \qquad m = v, b, \forall i \in \mathcal{I}^m \tag{9}$$

$$Z_{i,d_i^m}^m - U_i^m = 0 \qquad m = r, \forall i \in \mathcal{I}^m \tag{10}$$

Constraint (8) formulates the mass balance constraints for arriving, executed and delayed tasks for each vehicle. It also ensures that no container movement requests are executed prior to the arrival of a vehicle. Constraint (9) imposes that each vessel and each barge is completely served prior to its planned departure. Constraint (10) determines the number of non-executed tasks at the departure of a train.

$$S_{i,t}^m \geq \frac{p_i^m - Z_{i,t}^m}{p_i^m} \qquad m = v, \forall i \in \mathcal{I}^m, t = r_i^m, \ldots, T \tag{11}$$

$$X_{i,t}^m \leq X_{i,t-1}^m - k_t \cdot (S_{i,t}^m - 1) \qquad m = v, \forall i \in \mathcal{I}^m, t = r_i^m + 1, \ldots, T \tag{12}$$

The service of a vessel requires some preparation and coordination. Therefore, no additional straddle carriers are allocated to a vessel once its service has started, but retrieving superfluous straddle carriers is possible. Constraint (11) checks if the service of a vessel has been started prior to period *t* (*t* excluded). Constraint (12) allows the allocation of any number of straddle carriers to start serving a vessel. Once the operations have started it is only possible to withdraw allocated straddle carriers.

$$Y_{i,t}^m \geq \frac{Z_{i,t}^m}{p_i^m} \qquad m = b, \forall i \in \mathcal{I}^m, t = r_i^m, \ldots, d_i^m \tag{13}$$

Each period a barge spends at the terminal is penalized. Constraint (13) and the objective function determine for each period if a barge has been completely served and may leave the terminal or not.

$$X_{i,t}^m \in \mathbb{N}^+ \qquad m = v, b, \forall i \in \mathcal{I}^m, t \in \mathcal{T} \tag{14}$$

$$X_{i,t}^m \in \mathbb{R}^+ \qquad m = r, \forall i \in \mathcal{I}^m, t \in \mathcal{T} \tag{15}$$

$$\sum_{i=1}^{I^m} X_{i,t}^m \leq X_t^m \qquad m = r, \forall t \in \mathcal{T} \tag{16}$$

$$X_t^m \in \mathbb{N}^+ \qquad m = r, c, \forall t \in \mathcal{T} \tag{17}$$

$$\sum_{m=v,b} \sum_{i=1}^{I^m} X_{i,t}^m + X_t^r + X_t^c \leq k_t \qquad \forall t \in \mathcal{T} \tag{18}$$

Straddle carriers may either be allocated to one vehicle or shared among all vehicles of the same transport mode. Constraint (14) imposes that straddle carriers are allocated to exactly one vessel and one barge per period by preventing a partial allocation of straddle carrier to vehicles. Constraint (15) makes resource sharing for trains possible by allowing a partial allocation of straddle carriers. In order to prevent from resource sharing with other transport modes, constraints (16) and (17) are added. Constraint (17) also allows resource sharing among trucks and prevents resource sharing between trucks and other transport modes. Constraint (18) guarantees that the total number of allocated straddle carriers does not exceed the number of straddle carriers available at the terminal at each period.

3 EXPERIMENTS AND RESULTS

We estimate the impacts of an appointment system on the service quality of trucks and on trains, barges and vessels. In 3.1, we compare results of the optimization model for a terminal with an appointment system presented in this paper with the results for a terminal without appointment system presented in Zehendner *et al* (2011). Without the appointment system, trucks are served in FIFO order and each period they spend in the terminal is penalized. Service strategies for trains, barges and vessels are identical. In 3.2 the results of the optimization model are validated in a stochastic environment via discrete-event simulation.

Experiments are conducted on historic data of one container terminal of the Grand Port Maritime de Marseille. Input parameters were obtained by analyzing data on container movements at the terminal and via discussion with the terminal operator. Ten instances (each one representing one working day) were created from this data. Those instances differ with regard to the number of vehicles (1 or 2 vessels, 0 or 1 barges, 0 or 1 trains, 279-769 trucks), their volumes and their time windows.

3.1 Optimization model

Each instance is solved for 14, 12 and 10 available straddle carriers for the model with an appointment system (maximal deviations of $\Delta = 1$ and $\Delta = 2$) and for the model without an appointment system. We chose the highest delay costs for barges, medium delay costs for trains and the smallest delay costs for trucks. All models were solved with IBM ILOG CPLEX 12. All feasible instances are solved in less than one second. Infeasibility is also discovered in less than one second.

Table 1 shows the delays of trucks, trains and barges for each instance for 10 available straddle carriers. Instances with different results for the different models are marked in bold. For the terminal without an appointment system, truck delays are measured as the number of periods trucks have to spend in the terminal. For the terminal with an appointment system, truck delays are measured as the deviation between preferred and assigned arrivals. In both cases, delays of barges are measured as the number of periods spent at the terminal and delays of trains as the number of not executed tasks. Vessels are not represented since planning the delay of a vessel is not possible.

If the number of available straddle carriers is not sufficient to serve all tasks the instance is infeasible. Without an appointment system, trucks can be delayed until the end of the working day. With an appointment system, the arrival of trucks can only be forwarded or postponed by Δ periods and trucks have to be served within the period to which they are assigned. This explains why instance 9 is only infeasible for $\Delta = 1$.

Truck delays decrease with the introduction of a truck appointment system. This suggests that the introduction of a truck appointment system reduces the time trucks have to spend at the terminal. Delays for a deviation of two are smaller than delays for a deviation of one since more scenarios become possible. Without an appointment system, truck delays represent waiting times at the terminal. With an appointment system, truck delays represent the deviation from preferred arrivals. Probably, deviations are less disturbing than waiting times at the terminal. Delays of barges decrease for instances 2 and 9 and delays of trains increase for instance 9. For the given priorities, truck arrivals are also shifted to free capacity for barges. Train tasks are the only tasks that may remain unexecuted at the end of the day. In this case, trains are delayed as the instance would be infeasible otherwise.

Table 1. Delays for 10 available straddle carriers

	Instance	1	2	3	4	5	6	7	8	9	10
Truck	w/o	0	**275**	**247**	**26**	0	**115**	**129**	inf	**607**	inf
	Δ = 1	0	**214**	**165**	**25**	0	**48**	**66**	inf	**inf**	inf
	Δ = 2	0	**204**	**139**	**25**	0	**48**	**66**	inf	**598**	inf
Train	w/o	0	1	-	0	0	0	0	inf	**2**	inf
	Δ = 1	0	1	-	0	0	0	0	inf	**inf**	inf
	Δ = 2	0	1	-	0	0	0	0	inf	**9**	inf
Barge	w/o	-	**6**	-	6	4	-	2	inf	**5**	inf
	Δ = 1	-	**4**	-	6	4	-	2	inf	**inf**	inf
	Δ = 2	-	**4**	-	6	4	-	2	inf	**3**	inf

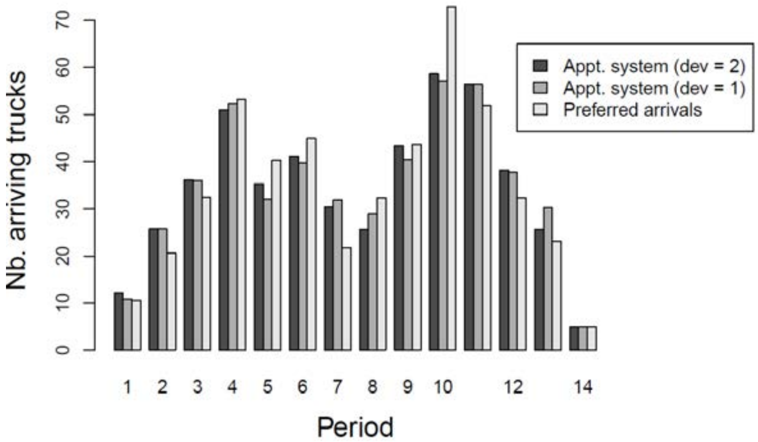


Figure 4. Preferred truck arrivals and proposed appointments

For 12 and 14 available straddle carriers, delays are almost identical for all optimization models since we aim to minimize the deviation between assigned time windows and preferred arrival times. Therefore, these results are not indicated here.

Our objective is to offer appointments close to the preferred arrival of trucks. To see if that objective is attained we compare the average arrival of trucks per period with the average number of appointments offered per period. Figure 4 represents the preferred arrivals of trucks and the appointments proposed by the appointment system with a maximum deviation of one and two periods. Only instances where truck arrivals are changed are presented. We see that the number of proposed appointments is close to the preferred arrival pattern of trucks. With an appointment system, fewer trucks arrive for periods 4 to 6. In fact, trucks are forwarded and postponed to free resources for barges which arrive at period 4. We also observe that the peak arrival at period 10 is smoothened out.

3.2 Simulation model

The optimization model assumes deterministic and known data which is clearly not the case at a real terminal. We use the discrete-event simulation model presented by Rodriguez-Verjan and Dauzère-Pérès (2010) to validate the findings of the optimization models in a stochastic environment. The simulation uses the same input data as the optimization model and the allocation proposed by the optimization model. Arrival times and volumes vary around the historic data. Three scenarios are analyzed for the handling and travelling times of straddle carriers with +/- 10%, +/-30% and +/-50% variation around average values. Simulation is only run for instances with different delays for terminals with and without appointment system. 1000 replications are executed for each instance.

Table 2 shows the performance indicators for trucks, trains, barges and vessels with and without a truck appointment system with a maximum deviation of two. For trucks, the average service time in minutes is indicated. For trains, barges and vessels the average number of not executed containers at the end of the day are indicated.
The appointment system reduces the average truck service time and the standard deviation considerably. Results of Student's t-tests show that differences are statistically significant for all variation levels with p-values of 0.015, 0.012 and 0.008, respectively.

For trains, the number of not executed tasks increases if the appointment system is used. Remember that the optimization model proposes an allocation with higher delays for trains for instance 9. This increase also appears in the simulation model. Results of Student's t-tests show that the differences are not statistically significant. The corresponding p-values are 0.455, 0.453 and 0.492, respectively.

Table 2. Simulation results for different levels of variations

	10% variation		30% variation		50% variation	
	w/o	w	w/o	w	w/o	w
Truck service time [min]	32.86	15.64	34.82	16.00	40.33	18.30
Std. dev.	27.79	8.86	28.94	9.12	31.75	10.22
Train not served [cont.]	0.21	0.96	0.21	0.96	0.35	1.09
Std. dev.	0.25	0.44	0.34	0.46	0.60	0.57
Barge not served [cont.]	0.00	0.07	0.01	0.13	0.24	0.68
Std. dev.	0.05	0.30	0.15	0.48	0.85	1.29
Vessel not served [cont.]	2.92	3.49	6.22	6.96	17.63	18.63
Std. dev.	5.38	5.66	7.67	7.89	10.37	10.50

The numbers of not executed containers for barges are very similar for both cases, but seem to increase slightly if the appointment system is used. The differences result from different resource allocations proposed by the optimization model. With an appointment system, the service of all barges starts in the period in which they arrive. Since we have small variations in the arrival of barges allocated capacity may remain unused at the beginning of the period. Results of Student's t-tests show that the differences are not statistically significant. The corresponding p-values are 0.261, 0.191 and 0.165, respectively.

For vessels, the numbers of not executed containers are very similar for both models. Differences result from different resource allocation for vessels. The total number of allocated resources is identical, but their distributions over time are different. It may thus be beneficial to get some more insight on the impacts of the allocation pattern on the resulting delays and to include these findings in the optimization model. Results of Student's t-tests show that the differences are not statistically significant. The corresponding p-values are 0.485, 0.771 and 0.813, respectively.

4 CONCLUSION AND OUTLOOK

In this paper, we extended a mixed integer linear programming model for the resource allocation problem to the case of a terminal operating with a truck appointment system. The model determines simultaneously how many appointments should be offered to trucks and how many straddle carriers should be allocated to trucks, trains, barges and vessels. This approach enables us to analyze the impacts of a truck appointment system on the service quality of trucks, but also of trains, barges and vessels. Results obtained by the optimization model and via discrete-event simulation show that a truck appointment system reduces truck service times and that it is also beneficial for other transport modes since trucks may be shifted to less crowded periods.

Our model assumes a linear relation between the number of allocated straddle carriers and the number of tasks that may be executed. In reality, the number of tasks executed per straddle carrier per period is influenced by several parameters: the arrival pattern of trucks within the period, the storage location of the container, if the transport of an import container can be combined with an export container or congestion. Our work may be continued by combining the deterministic optimization model with simulation or queuing models to include a nonlinear relation between the number of allocated resources and the number of executed tasks. Another possibility is to analyze the impacts of an appointment system for barges.

REFERENCES

Alessandri, A., Cervellera, C., Cuneo, M. and Gaggero, M. (2008), Nonlinear predictive control for the management of container flows in maritime intermodal terminals, *Proceedings 47th IEEE Conference on Decision and Control*, Cancún, Mexico, December 9-11, pp. 2800-2805.

Chen, X., Zhou, X. and List, G. F. (2011), Using time-varying tolls to optimize truck arrivals at ports, *Transportation Research Part E*, **47**, 1-18.

Gambardella, L. M., Mastrolilli, M., Rizzoli, A. E. and Zaffalon, M. (2001), An optimization methodology for intermodal terminal management, *Journal of Intelligent Manufacturing,* **12**, 521-534.

Giuliano, G. and O'Brien, T. (2007), Reducing port-related truck emissions: The terminal gate appointment system at the Ports of Los Angeles and Long Beach, *Transportation Research Part D*, **12**, 460-473.

Guan, C. Q. and Liu, R. R. (2009), Modeling gate congestion of marine container terminals, truck waiting cost and optimization, *Transportation Research Record 2100*, 58-67.

Huynh, N. and Walton, C. M. (2008), Robust scheduling of truck arrivals at marine container terminals, *Journal of Transportation Engineering*, **134**, 347-353.

Kang, S., Medina, J. C. and Ouyang, Y. (2008), Optimal operations of transportation fleet for unloading activities at container ports, *Transportation Research Part B,* **42**, 970-984.

Kozan, E. (2000), Optimising container transfers at multimodal terminals, *Mathematical and Computer Modelling*, **31**, 235-243.

Maguire, A., Ivey, S., Golias, M. and Lipinsk, M. (2010), Relieving congestion at intermodal marine container terminals: review of tactical/operational strategies, *Annual Meeting of the Transportation Research Board*, Washington DC, USA.

Morais, P. and Lord, E. (2006), Terminal appointment system study, *Transportation Development Centre of Transport Canada.*

Murty, K. G., Wan, Y.-w., Liu, J., Tseng, M. M., Leung, E., Lai, K.-K. and Chiu, H. W. C., (2005), Hongkong International Terminals gains elastic capacity using a data-intensive decision-support system, *Interfaces*, **35**, 61-75.

Rodriguez-Verjan and Dauzère-Pérès, S. (2010), Optimisation des opérations en logistique portuaire, *Master thesis in business engineering at Ecole des Mines de Saint-Etienne.*

Srour, J., Kennedy, J., Jensen, M. and Mitchell, C. (2003), Freight Information Real-time System for Transport (FIRST)- Evaluation final report, *U.S. Department of Transportation.*

Vis, I. F., de Koster, R. and Savelsbergh, M. W. P. (2005), Minimum vehicle fleet size under time-window constraints at a container terminal, *Transportation Science*, **39**, 249-260.

Zehendner, E., Absi, N., Feillet, D. and Dauzère-Pérès, S. (2011), Solving the resource allocation problem in a multimodal container terminal as a network flow problem, *Proceedings 2nd International Conference on Computational Logistics*, Hamburg, Germany, September 19-22, pp. 341-353.

TESTING, TUNING AND TRAINING TERMINAL OPERATING SYSTEMS: A MODERN APPROACH

C.A. Boer[1] and Y.A. Saanen[1]

[1]*TBA BV, Karrepad 2a, 2623AP, Delft, The Netherlands, csaba.boer@tba.nl, yvo.saanen@tba.nl*

ABSTRACT

A terminal operating system (TOS) is essential for efficient and productive terminal operations, as it supports planning, scheduling and equipment control. Increasingly, functions in the TOS are automated. As scope and level of automation are increasing, it is crucial that the software is well tested and fine-tuned before putting it in live operation. The traditional ways of testing and tuning the terminal operating system, as well as training its users are limited, leading to unnecessary risks. In this paper, we present a meanwhile proven, safe and inexpensive approach to test and tune the terminal operating system and train operators on an emulated virtual terminal. This novel approach in the field of container terminals has been successfully applied at over twenty container terminals during the previous past years.

Keywords: simulation, emulation, container terminal, optimization

1 INTRODUCTION

A terminal operating system (TOS) is a software application supporting the planning, scheduling and equipment control activities of a container terminal and by this being responsible for accurate operations within the terminal (Agerschou *et al*, 2004), (Stahlbock and Voß, 2008). As such, it is the heart of terminal operations, making its reliability and ability to enable high performing operations of essence. Even short hick-ups can cause substantial financial damage to a terminal as one hour of operational downtime may cost in excess of $50,000 for a large container terminal.

At the same time, terminal operators are requesting more functionality from their TOS, not in the last place while they are searching for greater process automation, going away from traditional manual planning and dispatching practices. This added functionality leads to further complication of the software in itself, adding to the risk of instability and thus operational downtime. Therefore, it is not surprising that, despite the great effort of software providers and - effort of future users, today’s practice is that TOS implementations and upgrades can cause quite some limitations and performance issues. One of the problems we observed, is the way the systems are tested and commissioned is limited to isolated processes, whereas day-to-day practice involves all kind of real-time interactions between the system’s components and processes executed. Besides, we noticed that during the testing phase, there is little attention for the performance that the TOS enables. If scheduling and dispatching algorithms are not clever enough, performance may be limited by the control software, rather than by the physical equipment executing instructions.

As the last points become of eminent importance, they motivated us to investigate the possibility to consider a simulated container terminal environment and a real TOS allowing for off-line experimentation with the TOS under near-to-live scenarios, hence including nearly all relevant dynamic interactions and events that are experienced during live operations at a container terminal. Herewith a clear overview can be created in which parameters can be fine-tuned during software development, commissioning, and during training and operations. As we could not find anything suitable in the market place at the time we initiated this activities, we developed an emulation tool – called CONTROLS (which stands for CONtainer TeRminal Optimised Logistics Simulation). In the meantime two other systems appeared in the market used for testing TOS using emulation: VITO (Virtual Terminal Optimization) and ChessCon from Institute of Shipping Economics and Logistics (Schütt, 2011).

This paper concerns the concept behind emulation, as well as its application to test, tune and train various TOS's. In the first place we present the concept of emulation in general, followed by the specifics of emulation for container terminal. After that we discuss in more detail the added value and application of emulation for testing, tuning and training TOS's. During the discussion we present some findings from several projects that have been carried out using emulation as a modern approach to perform testing and tuning the TOS, as well as training its (future) users. In the last section, we draw conclusions and provide an outlook for future development.

2 EMULATION OF CONTAINER TERMINAL

Emulation refers to the ability of a software application or physical device to imitate another software application or device. Emulation is also considered as a certain phase within testing process of a controlled system. The development of a complex system (e.g. a container terminal), which is controlled by a separate control system (e.g. a TOS), may include one or more of the following phases, which aim to test the system during different design stages (Auinger *et al*, 1999).

- Full simulation: includes the simulation of both the complex system and the control system;
- Real-time control: uses real complex system and simulates the control system;
- Emulation: simulates the complex system and uses real control system;
- Prototyping: involves tests with real complex system and real control system.

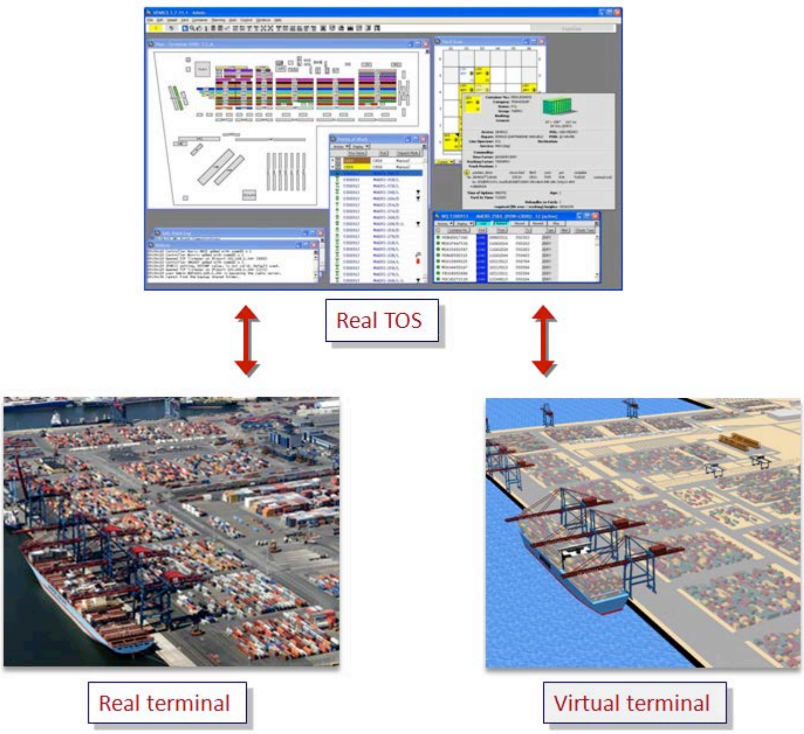


Figure 1. Prototyping vs. Emulation

While full prototyping seems the most realistic testing possibility, it is quite expensive to build and experiment with the whole prototype system, especially because it involves the risk of errors if the possibilities of its design are not tested thoroughly beforehand. Full simulation involves lower costs, however, it may disregard some phenomena that are present in the real system, or it may contain additional factors that might influence the outcomes. Emulation and real-time control have the advantage in that they can be carried out in a cheaper way than full prototyping, they stay closer to reality, and they are, therefore, less time-consuming than full simulation (Verbraeck *et al*, 2000), (Mueller, 2001), (Boer, 2005).

An emulation of a container terminal (a virtual representation of the container terminal's physical processes, and related operator behaviour) acts as a real terminal, i.e. provides a valid representation of the physical processes at a terminal (equipment behaviour, driver behaviour, operational scenarios – gate arrivals, train arrivals, vessel arrivals) (Schütt, 2011). This can be linked to the TOS in such a way that the TOS treats the model as the "real world" (via existing interface between TOS and equipment), and it can be used to run operational scenarios, either as they occurred in the past or as configured by the user (see Figure 1).

By examining the performance of the terminal controlled by TOS under various emulated conditions, an assessment can be made of the system (TOS but also the operational processes) in its actual configuration. This assessment is made by actually running an operation as it would normally take place at a terminal as well, however, without moving containers in a physical way. Now, the TOS communicates with virtual machines, drivers, clerks and other peripheral systems, rather than real ones. The communication protocol is completely identical to the communication in real-life. The emulation model includes a representation of all relevant processes at the terminal, e.g. the lay-out (yard, rail terminal and quay cranes), a model of the equipment (kinematics, driver behaviour, routing, disturbances, and availability), and performance measurement functionalities. All performance relevant interactions that take place between the equipment at the terminal (including the gate and rail terminal) have to be defined and supported by the interface between the TOS and the emulation tool (Kassl *et al*, 2008).

The performance of a terminal is jointly determined by the proper functioning of the TOS and the ability of the operators to use it. Given these aspects, we decided to develop an emulation environment with the aim to provide support by:

- Allowing for realistic, comprehensive, safe and inexpensive testing of the TOS for both green-field sites and for existing sites, implementing a new TOS, or receiving an upgrade.
- Enabling off-line tuning of TOS parameters, by allowing forecasting simulation, replay animation, and long term projection simulation.
- Allowing for realistic, real-time training, in a completely operational setting, for individual operational staff or complete teams (so-called human in the loop).

In order to enable emulation for the above mentioned goals, testing, tuning and training, we developed the CONTROLS emulation tool (Boer and Saanen, 2008). CONTROLS supports various types of operations and various TOS's. The tool consists of a number of modules that can be applied and configured for each terminal. Experimenting with CONTROLS typically involves three steps. The first step, is the definition of experiment scenarios. The second step is the execution of the scenarios in the combined TOS and emulated terminal environment. The last step is to analyse the results of the experiments for each scenario. In the next sections results of various case studies are presented, illustrating the added value of emulation approach using CONTROLS for testing, tuning and training terminal operating systems.

3 TESTING USING EMULATION

In order to ensure proper working of a TOS, vendors as well as terminals spend a huge amount of time testing it. The way this is done, we address as “traditional software testing”. Although, this is in many cases highly automated, it is typically limited to testing specific processes in isolation, and just running particular scripts. In addition, there is still quite a lot of manual testing going on.

Table 1. Traditional software testing vs. Testing using dynamic emulation

	Traditional software testing	Testing using dynamic emulation
Scope of testing	Pre-programmed test scripts Always the same	Complete operations Dynamic & comprehensive Terminal at capacity
Way of testing	Isolated components or business processes	Complete operations, with all interactions
Way of assessment	Test successful	Test successful Performance achieved

Emulation does not replace traditional ways of software testing, but adds on, for complex testing (Auinger *et al*, 1999). In Table 1, the differences are listed. When a testing exercise is executed with emulation, the control software is always in the loop. Test experiments typically comprise of an entire day of operations, and cover all kind of processes. From obvious loading and discharge, to more complex operations such as twin-lift, tandem-lift and dual cycling. The full chain of processes between gate, rail and vessel are covered. Experiments can be repeated as long as it takes to complete them flawless, and with the required performance, both technically (i.e. response times), and functionally (productivity and service levels). In a typical case, 5 – 10 typical operations are stored (‘scenarios’), and used as set of experiments that are representative for the (existing or future) operation.

Past experiences clearly show that many bugs are still found after the regular test cycles of the TOS vendors. We have also the experience that complex bugs are more likely to be found using emulation, as the coverage of the test scenarios is much greater. Furthermore, emulation-based testing automates quite a substantial amount of manual test cases. With prudent upfront mapping of test cases to the emulation scenarios, it can save 40 – 60% of manual (repetitive) testing.

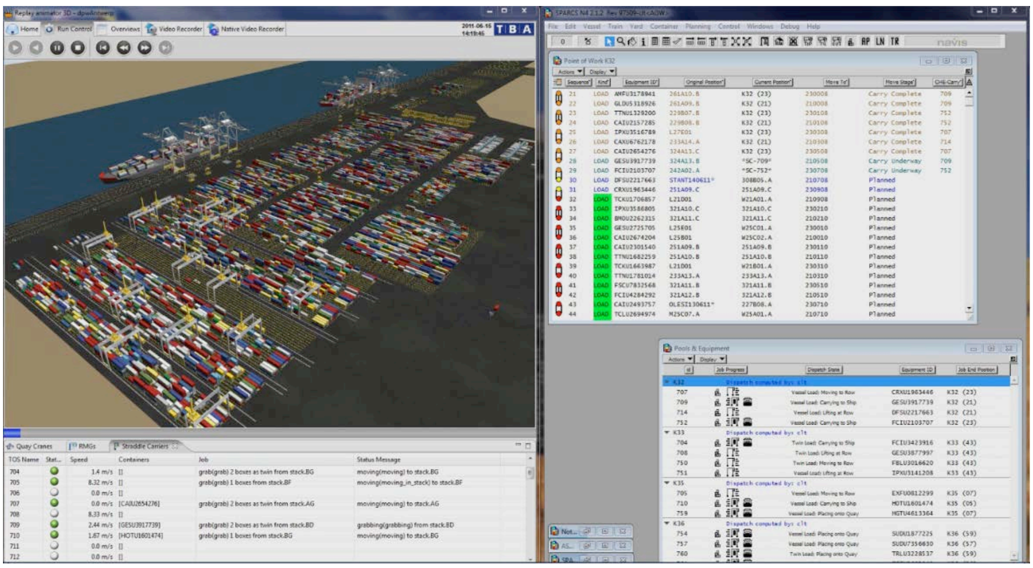


Figure 2. RMG terminal in CONTROLS side by side to the TOS (SPARCS N4)

In emulation projects using CONTROLS, the emulation tool has been mostly used to support the introduction of new TOS or to update an existing one. In these projects, emulation proved to be a very valuable tool, enabling the development team to detect and solve about 95% of the errors that typically are found *after* going live.

Furthermore, the visualisation accompanying the emulation, provided useful insights during the process (see Figure 2). While customers of a TOS usually have to wait until the system goes live, they were now able to operate a virtual operation long before the system went into real operation. This allowed future operators to be trained and also provided early feedback from the future users.

Finally, emulation enabled the team to stay focused on the performance of the terminal, or system. Where typically, the focus shifts from performance to “getting the things to work”, a continued focus could be kept on performance (think of quay crane productivity, equipment utilization, stacking efficiency, truck service, etc.) because CONTROLS allowed for this. In one particular instance, the final delivery by the TOS vendor only achieved about 60% of the required productivity level, as a result of errors, malfunctioning algorithms, and the need for fine-tuning parameters. The emulation allowed to detect the errors, and guide the fine tuning in a systematic way, with the resulting performance numbers as way to assess the impact of the solutions developed. Finally, the targeted performance was achieved, and later also reproduced in reality.

4 TUNING USING EMULATION

For basic installations of a TOS, the level of automation of decision-making by the TOS is limited. However, more and more, additional functionality is being added, to handle large-scale operations in which human decision-making becomes difficult, to increase the density of the yard (more throughput per hectare), and to improve equipment productivity. The modules that are being added typically contain a large number of parameters that are interrelated. Typically, the settings are configured and tested in live operations, which are by nature not suitable for finding satisfying or optimal configuration. Live operations are not repetitive, are highly affected by irregular events (e.g. break-downs) and human behaviours, and the measurement tools are also limited.

In order to tune the parameter settings and verify or optimize the algorithms, emulation can be used off-line using real operational scenarios. These scenarios are either real scenarios, meaning that it’s a replay of a past existing operation, or they are representative created scenarios, for instance for peak circumstances (e.g. full yard, 20% volume growth, or a full berth after a storm) or breakdown situations.

The execution of a scenario consists of an initialization step during which the user (one time) plans the operation. These plans (settings) can be stored, in order to be able to repeat this experiment without initializing it again. The duration of an experiment run depends on the TOS, the emulation tool is capable of running as long as the TOS controls the operation. Attended runs may last as long as the user wants, as the emulation behaves as the real world would behave. Unattended runs, however, depend on the capability of the TOS to run without an operator controlling it.

As such, a past operation can be run over and over again using various parameters, rules and equipment configurations, in order to determine what is best in particular situations. The more complex algorithms and software is deployed, and the more automation is being introduced (both from control perspective as well as equipment perspective), the more need for such an approach for emulation rises.

Through emulation the experiment runs may be started for various scenarios. The output of the experiment runs can be saved and is available for in-depth analysis and reporting tools. The TOS performance can be assessed by comparing the performance of various simulated operations, such as QC productivity, equipment productivity and status, truck and rail service times, number of shuffle moves, etc..

Emulation can be used either before implementing new software, focusing on the contribution to performance, or during operations, as continuous improvement tool. As circumstances change, parameter settings in the software may (and will) also need to change to accommodate the increased yard density, equipment availability or gate peaks. In the next two subsections we present two tuning cases when emulation (CONTROLS) provided support to improve the performance of the TOS.

4.1 Improve truck turn-around time

In two of the cases (so far), we applied the emulation to improve productivity in a live environment, replaying past operations, evaluating different parameter settings. In one case, we aimed at improving truck turn-around time by improving straddle carrier dispatching. It appeared from the experiments that the truck turn-around time during the peak – by just changing the dispatching algorithm and the decision factors – could be reduced by more than 30% on average, and the peaks by even 50%. At minimal investment – it implied some changes to the TOS – the service to trucks could be improved drastically. In other words, the number of straddle carriers required to handle a certain truck peak could be reduced, saving labour and equipment costs. Figure 3 depicts the turn time and straddle productivity from this project. The optimized settings determined using emulation were implemented in August 2010. Since then, turn time has been on averaging around 37 minutes, whereas 7 months before, they averaged at 49 minutes. This means a decrease of 25% of the turn time. At the same time, straddle productivity increased from average of 2.4 trucks per hour to 4.1 trucks / hour, an increase of 70% thus. Note that January and February 2011 were bad months due to excessive snow fall.

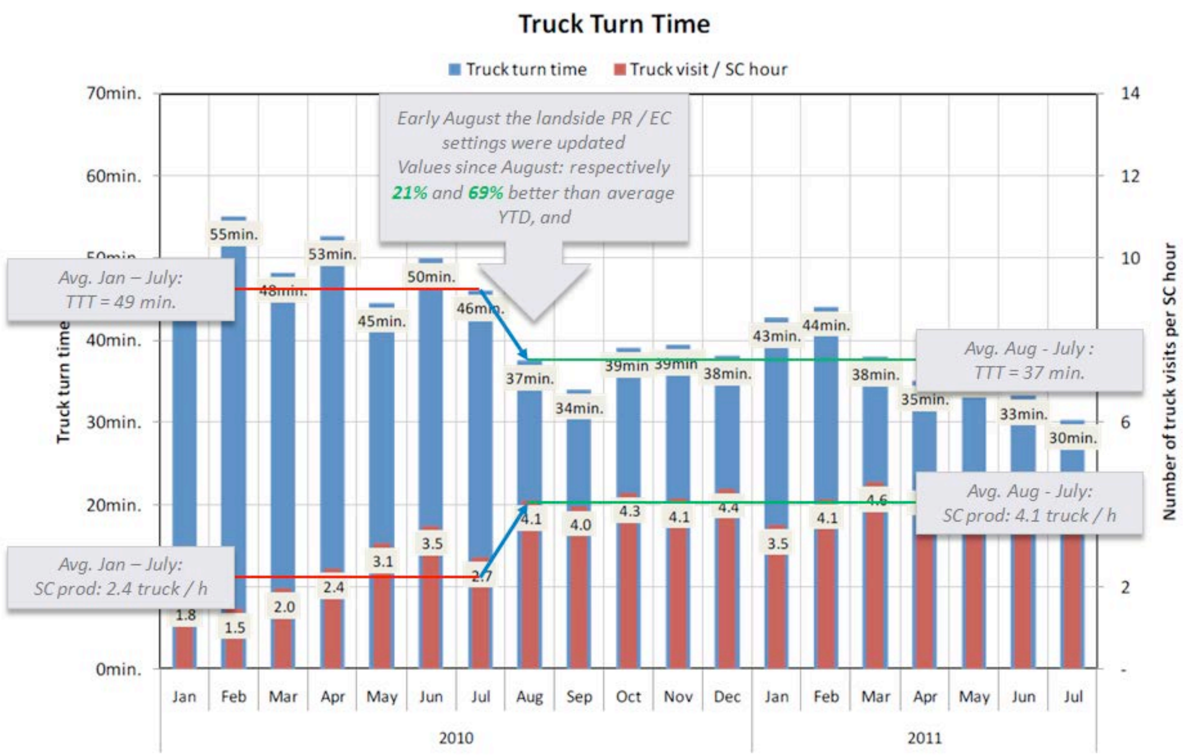


Figure 3. Turn time and straddle productivity (in trucks handled per hour)

4.2 Improve the yard planning strategy

Most recently, we carried out some studies for an RTG terminal that uses SPARCS terminal operating system. The goal of the study was to investigate the possibility to replace the currently applied yard planning strategy (based on the use of pre-stacks) with controlled random stacking strategy.

Proper yard planning strategies help to assign the containers to an optimal position in the yard. As a result of this, the re-handle moves and yard shifts can decrease, and the yard utilization and productivity can increase. In theory there exist different strategies, such as pre-marshalling (Chen, 1999), (Lee and Hsu, 2007), (Caserta *et al*, 2011), sort and store (Kim and Kim, 1999), (Kim and Park, 2003) controlled random strategy (Duinkerken *et al*, 2001) (Dekker *et al*, 2006), etc.

We defined different scenarios, in each one we modified the expert decking parameters according to certain aspects, such as: the workload of RTGs (e.g. increase/decrease the influence of RTG related variables), the travelling distance of TTs (e.g. increase/decrease the value of penalties related to terminal truck driving distance), specific yard settings (e.g. impossible to stack containers on top of containers that have a different type or which are planned to be moved). For each scenario, we carried out experiments and investigated which aspects are the most relevant. We concluded that with proper settings of the parameters the controlled random stacking strategy indeed can be a good choice as it improves the quay crane and RTG productivity.

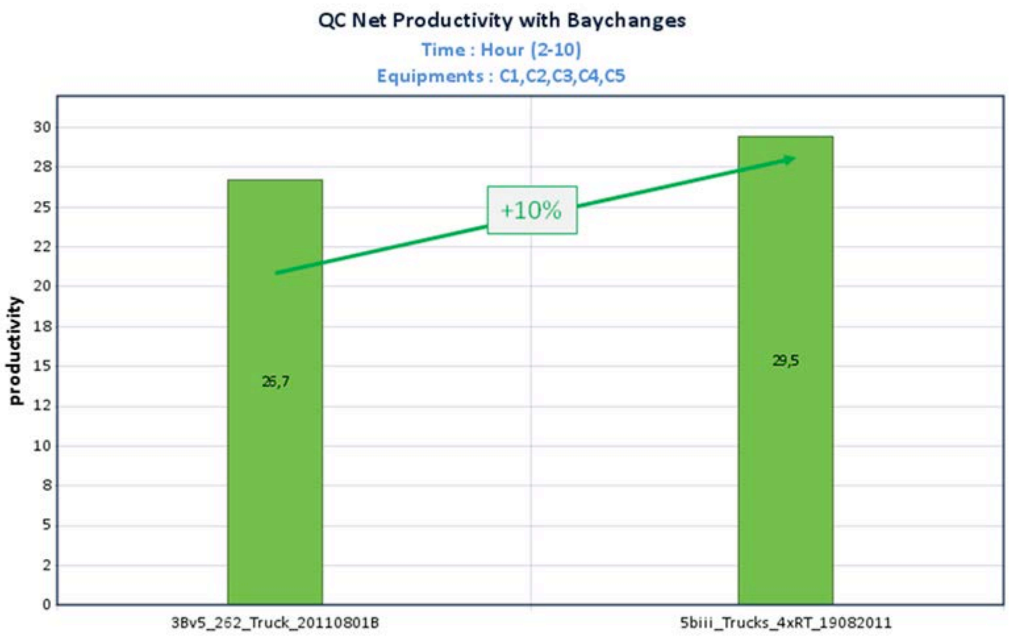


Figure 4. Comparing QC Net productivity applying different yard strategies

We achieved significant improvements (5-10% increase) of quay crane productivity applying the SPARCS expert decking functionality (see Figure 4). We realized this by changing the grounding parameters (for instance allocation filters in combination with equipment control parameters, the weight factor of travel distance, etc.).

5 TRAINING USING EMULATION

As operations become larger, denser and faster, and TOS contain more functionality to control these operations, the interaction between the TOS and the user (e.g. vessel planner, yard planner, equipment dispatcher) becomes more complex. This in the sense that there is more to decide for the user, the decisions have effects over long periods of time (up to weeks), and wrong decisions have a higher financial impact. Therefore, it's crucial that TOS operators (to address all its users) know how the TOS works, and how they should handle irregularities and events that the TOS cannot handle automatically. This starts from simple "fail to deck" events[1], reacting to breakdown events and rescheduling work, to vessels arriving off-window, and deciding where to berth them, so that the cascading effect on other vessels is minimal, whilst still allocating a reasonable berth to the vessel in question.

Training of TOS operators is typically performed on the job, which means that a newcomer sits besides an experienced person for some time, and then is deployed on its own. The likeliness of being capable then of handling less frequent but high impact events is small, which bares the risk of wrong decision-making. Besides, on the job training is typically not followed by any standardized form of evaluation to ensure that the newcomer is now able to handle the situations he will be faced with during operations. In addition, during his work, there is little evaluation of his work either, i.e. there is typically no feedback loop from work (e.g. planning) done in the past to the actual realized performance. As such, inadequate capabilities may remain unnoticed for longer periods of time.

In the emulation-based training, a real operation is being emulated in real-time, and the TOS user is fulfilling his role as part of the operational team. The TOS user fulfils his regular tasks, using the TOS, making his decisions, with the only difference to reality that his work leads to virtual container movements. Using emulation, "difficult" scenarios can be configured and events can be prepared (e.g. breakdowns, changing information, drivers that arrive in the wrong sequence) that test the capabilities of the TOS user. As such, the TOS user is trained in an environment that allows focusing on all tasks under regular and irregular circumstances, without affecting live operations.

An emulation-supported training that we have conducted, involving 6 vessel planners clearly shows the impact of a good plan. Each of the 6 planners was targeted to plan the same vessel. The vessel was consequently executed after completion of plan, all against an emulation of exactly the same operation, i.e. the same amount of equipment, the same initial yard, the same behaviour of equipment. The results are shown in Figure 5; it shows for each vessel planner the average crane productivity (in bx/h) as well as the vessel turn time (in hours). Note that a higher crane productivity does not always mean a shorter turn time, as work distribution among the cranes (up to 4 cranes were deployed per vessel) determines how long it takes to handle a vessel.

[1] The inability of the TOS to place a container fully automated in the storage yard, due to a lack of space within the user-defined allocation ranges.

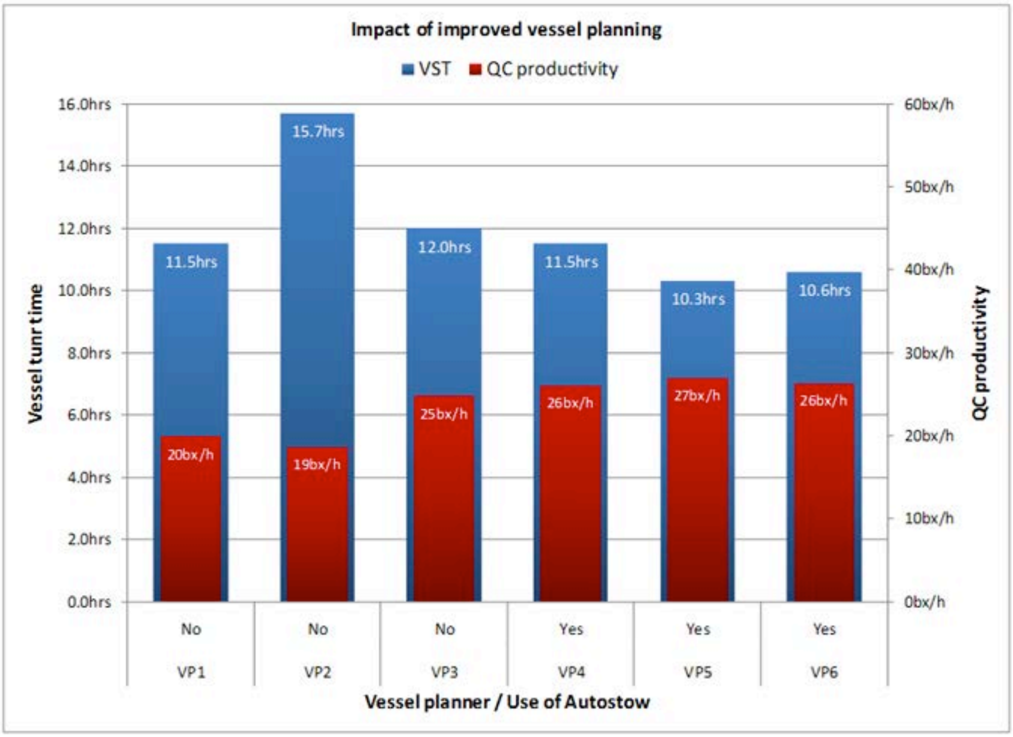


Figure 5. Results of vessel planning experiment

Three vessel planners were requested to use the provided automated stowage planning module available in the TOS. Despite their lacking experience with this tool, they all performed better by using it. Unfortunately, the group is too small to say anything statistically sound about the contribution of the automated stowage planning, however, the results as shown in Figure 5 give a clear indication that it is substantial. The average turn time (see the column in hours, varying from 10.3 hours till as high as 15.7 hours) decrease with 18%, whereas the average crane productivity increased with 26%. Furthermore, all planners that used the automated stowage planning turned the vessel quicker than the ones that practiced common procedures, and on top needed 25% less time to complete the planning process. Moreover, we can say that this way of training allows objective measurement, and safe try out of new methods, in this case for vessel planning. The case studies clearly show that the presented emulation approach indeed provides a safer and cheaper way to test and tweak the TOS and train operators on an emulated virtual terminal.

6 CONCLUSION

Due to the continuous volume increase, the need of handling more and more containers within given time window, and a limited yard place; larger vessels, new generation equipments, bigger terminals and more optimized operations are considered (Saanen, 2004). Since a container terminal's performance relies on its TOS, it is vital to continuously improve the capability of the TOS. As a TOS gets more advanced and as terminals get more reliant on their TOS, the importance of sophisticated test, tuning and tweaking tools gets more vivid. Simply allowing TOS vendors to deliver new releases without running realistic scenarios, installing TOS without detailed performance testing during the commissioning, and developing TOS without the use of tools that make the performance metrics explicit, is something that can entail a huge risk and expenses.

In order to reduce the risk and expenses, an emulation tool has been developed, called CONTROLS. Emulation allows the user to experiment with the real TOS without the risk of affecting (negatively) real operations. Problems caused by the TOS can be recognized immediately and solved before the software is put live. With CONTROLS, we have taken container terminal emulation to a mature point, offering a tool that can complement live operations with a valid, detailed, and advanced laboratory environment for testing, tuning and training. CONTROLS has proven itself in the various applications at the terminal – meanwhile we completed more than 20 projects -, by finding bugs, malfunctioning algorithms, allowing for tuning parameter settings as well as training operations staff in a systematic and measureable way.

The next challenge is to make this way of training the standard, as flight simulators are for pilots. So far the experience with the human in the loop, limited to three cases, and adoption goes slow. As this approach deviates substantially from today's way of training, using the emulation as training environment is a big step.

In addition, tuning algorithms and settings that are related to grounding containers, require experiments with a long duration. Here, today's TOS are not supportive in the sense that almost continuously (and at least every shift) user input is required to keep the operation – also in emulation mode – going. For tuning purposes, this is quite labour intensive, which is a potential bottleneck for extensive experimentation. We intend working on simulated TOS users to overcome that matter, but there is a long way to go, as the information that they use, is much richer than what the TOS contains, i.e. decisions of users are based on additional information as well (received by various media for instance).

Emulation allows not only experimenting with existing terminals but also with the future terminals. Accordingly, emulation also provides terminals with the opportunity to increase throughput without adding equipment or yard space: simply by *doing more with less*. As volumes increase, costs rise, and space is scarce, emulation provides a very promising way out.

ACKNOWLEDGEMENTS

We would like to thank our colleagues for their support in developing and testing CONTROLS.

REFERENCES

Agerschou, H., Dand, I., Sorensen, T., Ernst, T. *et al*. (2004). Planning and design of ports and maritime terminals. 2nd ed. Thomas Telford Ltd.: London

Auinger, F., Vorderwinkler, M. and Buchtela, G. (1999) Interface Driven Domain-Independent Modelling Architecture for "soft-commissioning" and "reality in the loop". *In Proceeding of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 798-805. Institute of Electrical and Electronics Engineers, Inc.: Piscataway, New Jersey.

Boer, C.A. (2005) *Distributed Simulation in Industry*. ERIM Ph.D. Series Research in Management: Rotterdam: The Netherlands. Available via <http://hdl.handle.net/1765/6925>

Boer, C.A. and Saanen, Y. (2008) CONTROLS: Emulation to Improve the Performance of Container Terminals. *In Proceeding of the 2008 Winter Simulation Conference*, eds. S. J. Mason, R. R. Hill, L. Monch, O. Rose, T. Jefferson, and J. W. Fowler, pp. 1094-1102. Institute of Electrical and Electronics Engineers, Inc.: Piscataway, New Jersey.

Caserta, M. Schwarze S., Voß, S. (2011) Container rehandling at maritime container terminals. In *Handbook of Terminal Planning,* ed. J.W. Böse, Springer: New York, pp. 247-269.

Chen, T. (1999) Yard operations in the container terminal - a study in the `unproductive moves'. *Maritime Policy & Management* **26(1)**: 27-38.

Dekker, R., Voogd, P. and van Asperen, E. (2006) Advanced methods for container stacking. *OR Spectrum* **28**: 563-586.

Duinkerken, M.B., Evers, J.J. and Ottjes, J.A. (2001) A simulation model for integrating quay transport and stacking policies in automated terminals. *In: Proceedings of the 15th European Simulation Multiconference* (ESM2001), SCS: Prague.

Kassl, H-D., Puchert, T. and Schütt, H. (2008) Optimising advanced container handling technologies via distributed simulation/emulation. In *Proceeding of the 11th International Workshop on Harbor, Maritime & Multimodal Logistics Modeling & Simulation*, ed. A. Bruzzone, F. Longo, Y. Merkuriev, G. Mirabelli and M. Piera, Italy, pp 67-71.

Kim, K.H. and Kim, H.B. (1999). Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics* **59**: 415-423.

Kim, K.H. and Park, K.T. (2003). A note on a dynamic space-allocation method for outbound containers. *European Journal of Operations Research* **148(1)**: 92-101

Lee, Y. and Hsun, N. (2007). An optimization model for the container pre-marshalling problem. *Computers & Operations Research* **34**: 3295-3313.

Mueller, G. (2001) Using emulation to reduce commissioning costs on a high speed bottling line. *In Proceeding of the 2001 Winter Simulation Conference*, ed. B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, Institute of Electrical and Electronics Engineers, Inc.: Piscataway, New Jersey.

Saanen, Y.A. (2004) *An Approach for Designing Robotized Marine Container Terminals*. Ph.D. thesis, Delft University of Technology: Delft, The Netherlands.

Schütt, H. (2011) Simulation technology in planning, implementation and operation of container terminals. In *Handbook of Terminal Planning*, ed. J.W. Böse, Springer: New York, pp. 103-116.

Stahlbock, R. and Voß, S. (2008). Operations research at container terminals - a literature update. *OR Spectrum* **30**: 1-52

Verbraeck, A., Valentin, E. and Saanen Y.A. (2000) Simulation as a Real-time Logistic Control System: AGV Control with Simple++. *The New Simulation in Production and Logistics – Prospects, Views and Attitudes*: Berlin.

OFF THE BEATEN TRACK WITH FREE RANGING AGVS

Mark B. Duinkerken[1], Jaap A. Ottjes[1] and Gabriel Lodewijks[1]

[1]*Faculty of Mechanical, Maritime and Materials Engineering, Delft University of technology, Mekelweg 2, 2628 CD, Delft, Netherlands*

ABSTRACT

Routing of Automated Guided Vehicles (AGVs) is the process of determining routes for a set of AGVs to fulfill their respective transportation jobs. Routing has a relatively large impact on the productivity and flexibility of AGV systems. In most existing systems, for instance on automated container terminals, AGVs use a map of predefined, fixed paths that are combined to obtain routes. However, theoretically free ranging capabilities of AGVs allows them to use the entire traveling area. In this paper an approach is proposed to improve the flexibility and capacity of AGV-systems by determining individual routes based on optimal control theory. The model describes AGVs as economic individuals, optimizing their costs. This approach leads to a new control method for AGVs, using their free ranging and positioning capacities. The developed routing algorithm dynamically determines trajectories that are optimized regarding arrival time while avoiding static obstacles and collisions with other AGVs.

Keywords: Automated Guided Vehicle Systems, Dynamic Routing

1 INTRODUCTION

Until now, industrial systems of AGVs are using a fixed guide path layout. This layout is sometimes virtual, thus not incorporated in the infrastructure any more. However, this approach has several disadvantages. First, a high capacity cannot be achieved by simply adding more AGVs to the system. It can be shown that bottlenecks will inevitably occur and limit the maximum throughput of the system [Saanen 2004]. Furthermore, flexibility is low, new paths and infrastructure cannot simply be added and even a shift in the workloads for origins and destinations can cause the system to perform worse. Then, the system is vulnerable for disruptions; breakdown of a single AGV can cause a blocking that stops the system partially or completely. Finally, the arrival time at destination is uncertain, depending on the number of vehicles in a specific area and other factors not known during the planning of a transport job.

Free ranging AGVs are expected to overcome the disadvantages of fixed guide path layouts and lead to higher system capacity, more flexibility, lower costs and lower emissions. Letting the AGV leave the fixed guide path and chose its own path introduces a whole new class of issues. The study of so-called free-ranging vehicles until now mostly focuses on single vehicles (most commonly mentioned 'mobile robots') [van Turenhout 1994]. The motion planning for these vehicles can be categorized into holonomic path planning and non-holonomic path planning; the latter taking into account the dynamics between vehicle wheels and the driving surface [Barraquand 1991]. However, most research is not studying the performance of a multi AGV system but only the individual vehicle.

Improving the performance of container terminals has been the subject of study for many projects [Stahlbock and Voss 2008]. Traditionally, the key performance indicator for a transport system is throughput, i.e. the number of completed transport moves per hour. An obvious way to improve production for a single AGV, given a maximum driving speed, is to reduce the length of the path between origin and destination. However, in a transport system, the timing of the transport vehicles and the loading/unloading equipment on both sides of the transport task must also be taken into account. This leads to the insight that accurate start and arrival times are also important. The paradigm shifts from ‘maximize throughput’ to ‘minimize variance in transport times at the same throughput'.

2 RESEARCH GOAL

With free ranging, the performance of a transport system in terms of capacity and reliability might be improved. For free ranging, the AGV controller must dynamically plan the motion between the current location and the destination of a transport job and realize the specified arrival time, when feasible. The control model must use the available information on the planned routes of other vehicles to plan a collision free trajectory. During execution, the controller is capable to handle incidents.

The aim of this study is to develop and evaluate a control concept that enables free ranging AGV systems. The following subgoals for this research can be distinguished:

- To increase the autonomy of vehicles and to decrease the dependency on central controllers
- To maximize capacity utilization of installed infrastructure and vehicles and to minimize aberrations between planning and execution of individual transport jobs
- To increase robustness, agility, scalability and safety of automated transport systems

This paper contains the description of a control model called DEFT, which is an acronym for Dynamic, Evasive Free-ranging Trajectories. The simulation results act as a 'proof of concept' for the DEFT method.

3 OPTIMAL CONTROL

This chapter describes the method used for the calculation of individual routes. The algorithm is based on the route choice methodology of NOMAD [Hoogendoorn e.a. 2002], a microscopic pedestrian behavioral model. The specific form of the system's state and cost function in this work allows for fast calculation of the optimal velocity trajectory that minimizes the costs over the planning horizon.

In system theory, the basic model consists of two essential parts: a dynamic system and a cost function that is additive over time. Consider a discrete time system with state x_k, input v_k and disturbance w_k, then the system has the form:

$$x_{k+1} = f_k(x_k, v_k, w_k) \tag{1}$$

where k represents the (discrete) time in this system and runs from k=0 to a finite horizon k=N.

The state x_k is a vector describing the current 'condition' of the system. The input v_k is a vector with parameters which are used to influence the future states of the system. The term w_k represents a random parameter, also called disturbance or noise. If w_k is not zero, the system is called stochastic. For the purpose of this research which is trajectory planning the inclusion of noise is not needed. For the remainder of this paper a deterministic system is assumed, thus w_k equals zero.

The function f_k describes how the next state x_{k+1} evolves from the current state, the input and the random noise. If f_k is independent of time k, then the system is called time-invariant.

Let L_k be the costs for time k, which can be the sum of multiple cost components and depends on the values of time k, state x_k and input v_k. The parameter Φ represents the final costs (also known as terminal costs) at time N and depends on the end state x_N of the system. Then the cost function J is given by:

$$J = \sum_{k=0}^{N-1} L_k\left(x_k, v_k\right) + \phi\left(x_N\right) \tag{2}$$

Instead of input, the term 'control' will be used for v_k in the following paragraphs. The controls v_k can be restricted to a set U_k of admissable controls (3). Admissable controls thus depend on both time k and state x_k.

$$v_k \in U_k(x_k) \tag{3}$$

Goal of optimization is to minimize the cost function J by determining a range of controls (v_0, v_1, ..., v_{N-1}). With open loop control, all the values v_k are determined at time k=0. In closed loop control, when determining the value of v_k, information on the value of the state x_k is assumed to be available.

The dynamic programming algoritm, which is used to optimize the costs in (2), is based on Bellman's optimization principle [Bertsekas 1995]. In order to present this principle first an additional definition is required. The so called value function $V_k(x_k)$ is defined as the costs associated with the optimal control from state x_k at time k to the end of the planning horizon, time k = N. The value function $V_k(x_k)$ can be seen as the optimal cost-to-go, given the time k and the state x_k. It is clear from the definition of the value function that $V_0(x_0)$ represents the optimal costs for the original problem in (1) and (2). It can also be seen that the value of $V_N(x_N)$ is defined by the final costs function Φ.

Bellman's principle states that the value function $V_k(x_k)$ can be solved backwards in time from time k=N-1 to time k=0 using equations (4) and (5):

$$V_N(x_N) = \phi(x_N) \tag{4}$$

$$V_k(x_k) = \min_{v_k \in U_k} \left\{ L_k(x_k, v_k) + V_{k+1}(f_k(x_k, v_k)) \right\} \tag{5}$$

where the minimization in the righthand term of eq. (5) is taken over the set of admissable controls U_k.

The optimal control function v_k is a policy {v_0, v_1, ..., v_{N-1}} that minimizes the cost function J in (2). Bellman's principle presents a sufficient condition which an optimal control v_k must satisfy. However, in its general form equation (5) can not be solved analytically.

4 AGV ROUTE PLANNING

4.1 Description

The AGV routing system is defined according to the characteristics given in the previous chapter. The state x_k is the position of the AGV in a two-dimensional space. The control v_k is the speed vector that is applied on the AGV. The cost function L is the sum of several cost components; each component is designed to discourage a specific form of undesired behaviour (for instance running into an obstacle). The cost function is further discussed in the next sections. The final costs Φ is used as a reward function in case the destination is reached at time T.

4.2 Cost definitions

The costs L occur between k=0 and k=N and are therefore mentioned the running costs. The running costs are the sum of multiple cost components L_k:

$$L(x_k, v_k) = \sum_m c_m . L_m\left(x_k, v_k\right) \tag{6}$$

The factors c_m determine the relative weight of each of the cost components L_m. Four cost components are defined, namely costs for speed of movement (L_1), time pressure (L_2), static obstacles (L_3) and other vehicles (L_4). The first cost component L_1 depends on the speed v_k; the other cost components only depend on the position x_k.

4.3 Speed of movement

L_1 expresses the aversion of driving at higher speeds and implicitly puts a penalty on acceleration and deceleration.

$$L_1(v_k) = \tfrac{1}{2}.\|v_k\|^2 = \tfrac{1}{2}.v_k{}'v_k \qquad (7)$$

This cost component results in keeping a constant speed, as low as possible, just enough to reach the destination at the end of the planning interval.

4.4 Time pressure

Running cost L_2 is used to put time pressure on an AGV, resulting in a preference for the fastest route in time.

$$L_2(x_k) = 1 \qquad (8)$$

The function is a simple constant. The total costs resulting from this component is equal to the total travel time. This will result in a preference for faster routes.

4.5 Static obstacles

The third component L_3 expresses the cost of driving close to an obstacle B_i. An obstacle is described by a polygon. The function $d(B_i, x_k)$ defines the shortest distance between the position x_k and the obstacle, where a_o and b_o are scaling parameters. The weight of this cost determines the obstacle avoidance characteristics of the route planning system.

$$L_3(x_k) = \sum_{i=1}^{\#obstacles} a_o.\exp\left(\frac{-d(B_i, x_k)}{b_o}\right) \qquad (9)$$

The parameter a_o determines the maximum absolute value of this cost. The parameter b_o determines the distance at which the influence of this cost will become apparent for the AGV.

4.6 Other vehicles

The component L_4 expresses the cost of driving close to another AGV_i. The function $d(AGV_i, x_k)$ defines the distance between the position x_k and AGV_i, where a_v and b_v are scaling parameters. The weight of this cost determines the obstacle avoidance characteristics of the route planning system.

$$L_4(x_k) = \sum_{i=1}^{\#AGVs} a_v.\exp\left(\frac{-d(AGV_i, x_k)}{b_v}\right) \qquad (10)$$

The costs L_4 is similar to L_3. Basically, other vehicles are modeled as moving obstacles. Scaling parameter a_v and b_v are chosen such as to take the geometry and maneuvering capability of the vehicles into account.

4.7 Terminal costs

The final cost Φ is the cost at the end of the planning period N. The purpose of the optimization is to calculate a speed function that drives the AGV towards the desired destination. Therefore, the cost $\Phi(x_N)$ is negative if x_N is the destination, and zero if x_N is not the destination. A negative cost Φ is in fact a 'reward' for reaching the destination.

4.8 Interpretation of the value function

The final cost Φ can be seen as a reward for reaching the destination. Optimal control results in a speed function v_k that minimizes the costs. The total costs for the optimal speed v_k can be a positive or a negative value.

An interpretation can be to see the final cost Φ as a 'travel budget'. The budget must be large enough to cover all the expenses during travel. Travel expenses are the sum of all the cost components. If minimization of the value function results in a positive value, it can be concluded that the trajectory is not 'worth driving', in other words, that replanning is suggested. However, ignoring these positive costs and use the found trajectory is also an option.

5 SOLUTION METHOD

The problem stated in (4) and (5) is solved following a numerical approach introduced by [Fleming and Soner 1992] and adapted for a pedestrian model by [Hoogendoorn and Bovy 2002]. This approach solves the so-called HJB equation, the continous time equivalent of (5). The HJB equation is valid for a wide range of optimal control problems, but in general an analytical solution does not exist. For specific characteristics of the system, the problem can be solved numerically, using both a discrete timestep as well as a discrete spatial step. In this paragraph, three specific conditions are mentioned which are used.

5.1 Deterministic control

In [Fleming and Soner 1992] a finite-difference scheme is described which results in an approximate solution of the HJB equation for a controlled Markov diffusion process, thus for a stochastic system. In this paper, the same approach is followed but under the assumption that AGV control is deterministic. Therefore, the mean and variance of the noise w_t equals zero. While this paper follows the approach of Fleming, the stochastic terms in the equations will be left out.

5.2 State evolution

In [Hoogendoorn and Bovy 2002], a pedestrian is modelled as a single point controlled by a speed vector. Analog to this approach, in DEFT the state x_t is a two-dimensional vector presenting the location of the AGV in an euclidean space. The control v_t is a two-dimensional vector with the speed components along the two axis. State evolution is given by:

$$\dot{x}_t = v_t \tag{11}$$

It is straightforward to describe motion with a system like (11) and a lot of research has been done on these systems. For this class of problems, extensive literature is available under the name 'calculus of variations' [Bertsekas1995]. In fact, calculus of variations has a long history and predates the development of optimal control theory.

5.3 Structure of the cost function

To solve the optimal control problem, a cost function L with a specific structure is defined. The cost function L has one component quadratic in speed v_t and all other cost components only related with the state x_t. This cost function is defined in (12):

$$L_t\left(x_t, v_t\right) = \frac{c_1}{2} . \| v_t \|^2 + \sum_{m=2}^{n} c_m L_m\left(x_t\right) \tag{12}$$

Essential for the approach is the fact that only the first cost term depends on v_t. The fact that the remaining part of the cost function L is a linear combination of the cost components

L_k is not required for application of this approach; it is sufficient that the remaining part does not depend on the control v_t.

5.4 Trajectory calculation and stability condition

Under the three conditions mentioned in the previous paragraphs, the value functions $V_k(x_k)$, starting at k=N-1 backwards to k=0. are calculated. This proces results in a set of N value functions. After these are obtained, the related trajectory can be determined. The optimal trajectory starts at position x_0 on time k=0. Using the value-function $V_0(x_0)$, the optimal speed-vector is perpendicular on the iso-value lines of the value function. Thus, the method to determine the optimal speed vector v_k can be seen as a steepest-descent method applied to the value function V_k at time k. This method assumes infinite acceleration and deceleration.

In [Fleming and Soner 1992] a relation is presented between the discrete timestep h and the spatial stepsize δ for the stochastic system:

$$h.\left(\sigma^2 + \delta.\| v_k \|\right) \le \delta^2 \qquad (13)$$

Given a spatial stepsize δ, if the timestep h is chosen according to condition (13), the stability of the approximation method is garantueed. Define v_{max} as the maximum speed of the AGV. It is clear that v_{max} bounds the size of v_k in (13). Furthermore, σ^2 equals zero, as stated in the first paragraph of this chapter. It can also be proven that if h approaches zero, the value function V_k converges to its continous time counterpart V_t.

6 EXPERIMENTAL RESULTS

6.1 Experimental setup

The DEFT model is tested in the simulation framework described in [Duinkerken 2009]. Three categories of experminents are conducted to test the control model for theoretical and practial use.

6.2 Terminal costs and speed of movement

The simplest test scenario is a single AGV, planning a trajectory from an origin to a destination, without any obstacles. Distance between origin and destination is 250 m, maximum speed is set at 5 m/s. Running costs when driving at maximum speed thus equals $0.5 * 50 * 5^2 = 625$. The only costs involved are the cost of speed (see equation (7)) and the terminal costs. Obviously, the terminal costs must be higher then the total cost of movement otherwise the AGV will not move and simply collect the terminal costs. In case the terminal costs exceed the running cost, the optimal trajectory will be a straight line from origin to destination, starting at t=0 and ending at t=T, with the lowest possible, constant speed.

In table 1 the results of this experiment can be found. The terminal costs are chosen to be double the cost of running at maximum speed, thus 1250. Time horizon T is varied between the minimum (distance / v_{max} = 50) and 100. It can be seen that the average speed of the AGV is lower if the timewindow is larger. Furthermore, the AGV arrives a few seconds earlier then needed, because of a somewhat higher average speed then required. This effect is probably caused by the numerical approximation; it is reduced when a smaller spatial step and timestep is choosen. See for example the speed profiles in figure 1. For a horizon of 100 sec, the expected average speed is 2.5 m/s; a smaller spatial stepsize results in a better approximation of the theoretical result.

Table 1. Average speed and arrival time for different horizons

Horizon T (sec)	Average speed (m/s)	Arrival time
50	5.0	50.0
60	4.5	55.9
70	3.8	65.9
80	3.3	75.7
90	2.9	85.5
100	2.6	95.3

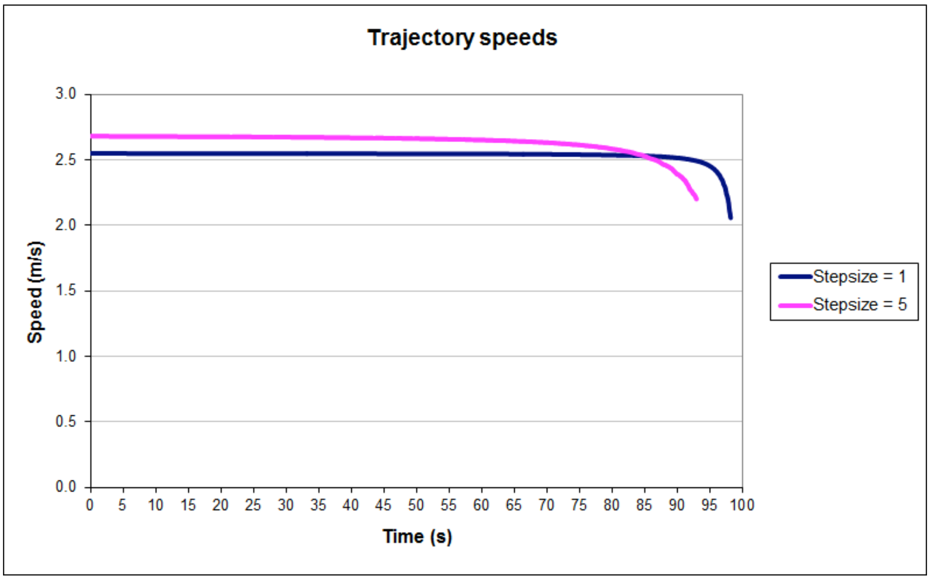


Figure 1. Speed profiles for different values of the spatial stepsize

6.3 Obstacle evasion

The next group of experiments introduces a single obstacle between origin and destination, introducing obstacle costs, see equation (9). This forces the AGV to divert from the straight line. Parameter a_o determines the absolute cost level of obstacles; in order for the AGV to leave the straight line, it must be higher then the costs for the detour. Detour costs are influenced by both the extra length of the trajectory and the eventual variation in speed. Parameter b_o influences the distance from which these costs are felt; it must be high enough to garantuee a safe distance between an AGV and an obstacle but not to high to prevent too long detours. The obstacle itself is a wall from 1 side of the terminal to the middle of the shortest path connection between origin and destination.

In table 2 the minimum passing distance between AGV and obstacle is presented for different values of a_o and b_o. In this example, both low and high values for a_o result in a detour avoiding the obstacle. The value for b_m has a high impact on the minimum distance between AGV and obstacle. Setting this value to high results in the inability of the method to determine a feasible trajectory within the given time horizon. Figure 2 shows the value function at time T/2; the trajectory is also shown in black for t=[0,T/2] and white for t=[T/2,T]. Green parts of the value function indicate that the destination can be reached (between t=T/2 and t=T) against costs lower than the terminal costs, yellow and red means that the running costs would be higher then the terminal costs.

Table 2. Minimum passing distances

Minimum passing distances (m)		b_o 2.5	5	10
	5	5.6	7.5	9.6
a_o	10	7.0	11.1	16.2
	200	15.3	25.8	no trajectory

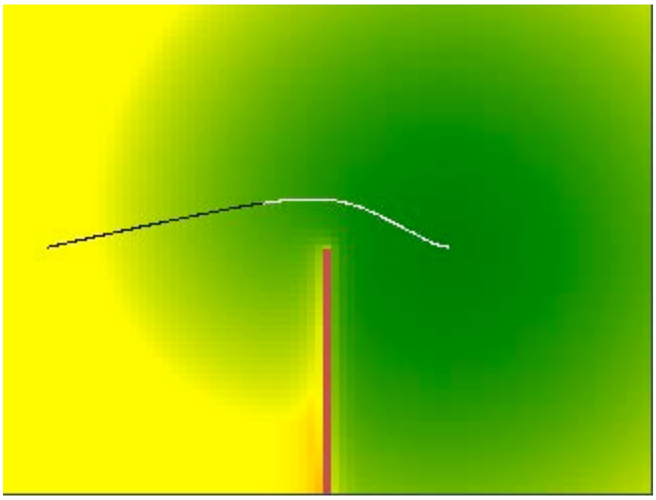


Figure 2. Trajectory and value function at t=T/2

6.4 Multi AGV system (running cost L_4)

The third experiment shows a single run of a terminal with 13 stacking cranes, 6 quay cranes and 25 AGVs over 3600 seconds. More details of the container terminal layout and experiment settings can be found in [Duinkerken 2009]. For this experiment a_v and b_v are chosen such that zero collisions occurred on the terminal. The number of completed transportjobs are given in table 3 and compared with other trajectory planning methods from [Duinkerken 2009].

Table 3. Terminal capacity for different routing methods

Layout	Method	Average waiting time	Average trajectory length	Average speed (excl. waiting)	Moves / hour
Cross-over	not safe	0.0	191.8	4.0	1861
Cross-over	Safety time = 0.1s	20.7	194.3	4.0	1288
Mesh	VRPTW	27.6	277.4	4.0	917
-	DEFT	0.0	216.5	3.4	1384

The first experiment, the not-safe cross-over variant calculates the theoretical maximum capacity. In this variant, the AGVs are colliding thus the resulting trajectories are not feasible. Implementing a safety time between AGVs result in a more realistic, much lower capacity. The mesh layout using a VRPTW control algorithm is an example of a realistic, fixed guide path layout [Möhring 2004]. As can be seen from table 3, the free ranging DEFT method promises a much higher terminal throughput. A screenshot of calculated trajectories can be seen in figure 3.

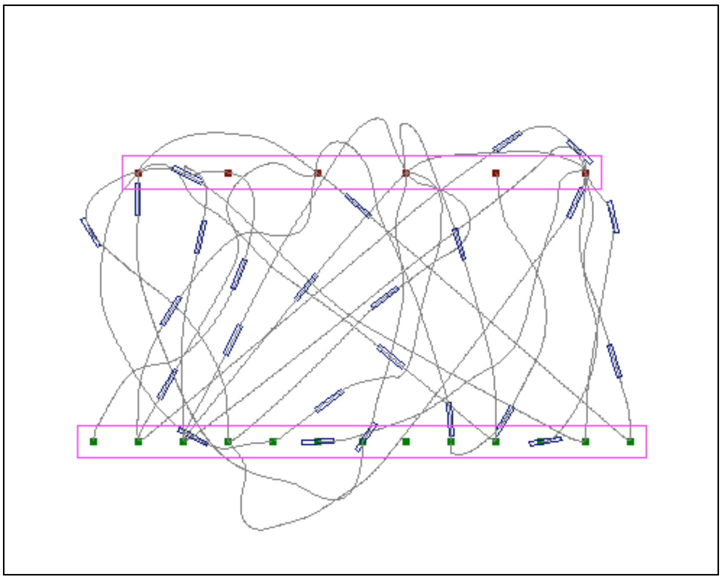


Figure 3. DEFT trajectories for 25 AGVs

7 CONCLUSIONS AND RECCOMENDATIONS

It can be concluded that the DEFT control method in potential does what is required: calculating dynamic, collision avoiding, free ranging trajectories for AGVs. Theoretically it results in an increase in terminal capacity, due to shorter paths between origins and destinations of transportjobs. Whether or not the research goals are reached must be studied further but the prelimenary results look promising. An operational controller is needed to implement the trajectories on the AGVs and resolve the infeasibility issues.

7.1 Reccomendations

A number of recommendations to further study the DEFT model can be given. Study of the feasibility of calculated trajectories is eneded and so is an evalution of the resulting AGV arrival times at destination. Then also a more realistic job dispatching approach can be tested.

Furthermore, there are many ideas for improvement of the model itself. First on this list is the search for a method to include a directional component in the model. AGVs are modelled as point-masses which is not very realistic. Also, a method to better deal with the non-holonomic constraints is required and can result in a better performance.

ACKNOWLEDGMENTS

This work was supported in part by the Next Generation Infrastructures Foundation.

REFERENCES

Barraquand, J. and J.C. Latombe (1991). Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. In IEEE International Conference on Robotics & Automation, 2328-2335.

Bertsekas, D.P (1995). Dynamic Programming and Optimal Control. Volume 1. Athena Scientific, Belmont, MA.

Duinkerken, M.B, J.A. Ottjes and G. Lodewijks (2009). Comparison of routing methods for AGVs. In: Proceedings of the Huntsville Simulation Conference HSC 2009. October 27-29, 2009. Huntsville, AL.

Fleming, W.H. and H. Mete Soner (1992). Controlled Markov Processes And Viscosity Solutions. Springer Science+Business Media, Inc.

Hoogendoorn, S.P. and P.H.L. Bovy (2002). Pedestrian route-choice and activity scheduling theory and methods. In: Transportation Research B. Vol. 38, No. 2, 169-190.
Möhring, R.H., E. Köhler, E. Gawrilow and B. Stenzel (2004). Conflict-free Real-time AGV routing. In: Operations Research Proceedings.
Saanen, Y.A. (2004). An approach for designing robotized marine container terminals. Ph.D. thesis, Delft. University of Technology, Delft, The Netherlands.
Stahlbock, R and S. Voß. (2008). Operations research at container terminals: a literature update. In: OR Spectrum. Vol. 30, Issue 1, 1-52.
Turennout, P. van (1994). Autonomous motion on wheels. Ph.D. thesis, Delft. University of Technology, Delft, The Netherlands.

YARD CRANE DISPATCHING TO MINIMIZE JOB TARDINESS IN CONTAINER TERMINALS

Shell Ying Huang[1], Xi Guo[1], Wen Jing Hsu[1] and Wei Lin Lim[1]

[1]*School of Computer Engineering, Nanyang Technological University, Singapore 639798, assyhuang@ntu.edu.sg, guox0006@ntu.edu.sg, hsu@ntu.edu.sg*

ABSTRACT

For container terminals to provide efficient services to its customers, one of the most important objectives in operation planning is to minimize vessel turnaround time. This means the vehicles have to reach the quay cranes such that they minimize the quay crane waiting time for them. This in turn means when vehicles go to yard blocks, they have deadlines for their loading/unloading jobs. The yard cranes should try to finish the vehicle jobs with minimal tardiness. Two provably optimal algorithms are presented to find the optimal YC job sequence for serving a fleet of vehicles for delivery and pickup jobs with scheduled deadlines and predicted vehicle arrival times. Some of these jobs require reshuffling of other containers. The objective is to minimize the total tardiness of incoming vehicle jobs. The first algorithm MTA* is based on the well-known A* search along with a proposed admissible heuristics. The second algorithm MT-RBA* is a recursive backtracking algorithm which uses the same admissible heuristic and a prioritized search order to accelerate the computation. Experimental results show that both algorithms significantly outperform the Earliest Due Date First and the Smallest Completion time Job First heuristics in minimizing job tardiness. When handling a planning window of 10 jobs, MT-RBA* only takes a few seconds to complete in most cases and half a minute in the very heavy workload cases.

Keywords: Yard crane dispatching, Optimization, Container Terminal

1 INTRODUCTION

Container terminals serve as crucial hubs in the transportation chain of the ever increasing global cargo flows. When a vessel berths at a terminal, a number of Quay Cranes (QCs) are allocated to serve the vessel. QCs first unload containers from the vessel onto in-terminal vehicles for transferring them to the container storage yard. A vehicle would take the containers to specific job locations at various yard blocks in the storage yard. Yard Cranes (YCs) pick up the containers from the vehicles and temporarily store it in the yard blocks. The operation of loading containers onto a vessel is carried out in the reverse order. From time to time external vehicles come into the storage yard to the yard blocks to get export containers unloaded or import containers loaded by YCs. In this paper, we assume that vehicles do not have unloading capabilities therefore they need the QCs and YCs to unload the containers from them.

One of the most important objectives of terminal operations is to reduce vessel turnaround time (Steenken *et al*. 2004). Previous studies have pointed out that YC operations are of great importance and likely to be a potential bottleneck to the overall terminal performance (Li *et al*. 2009). This is because when vehicles are delayed in the storage yard, they will not be able to reach their QCs on time. As a result, QCs' loading/unloading operations will be delayed and vessel turnaround time lengthened. In YC operation management there are two main problems: (i) deciding job sequence for an YC which we refer to as the YC dispatching problem; (ii) allocating YCs to different parts of the yard which we refer to as the YC deployment problem. We study the YC dispatching problem in this paper.

The YC dispatching problem was studied by Kim and Kim (1999) where they considered the loading operations only for a single YC with a given load plan and a given bay plan. A Mixed Integer Programming (MIP) model is proposed to minimize the total gantry time of the YC. Later, Kim and Kim (2003) and Kim *et al.* (2004) extended the study of this problem by comparing exact optimization, a beam search heuristic and a Genetic Algorithm (GA). Ng and Mak (2005) developed a heuristic to solve the single YC dispatching problem with different ready times to minimize the total job waiting time. It is known that for large problems, the MIP model has limited applicability due to the excessive computational times. On the other hand, heuristics cannot guarantee optimal solutions. Guo *et al.* (2011) applied A* search to compute optimal single YC dispatching sequence based on vehicle arrival times to minimize vehicle waiting times.

Jung and Kim (2006) considered 2 YCs working in one shared zone to support vessels loadings with a GA and a Simulated Annealing (SA) algorithm to minimize the make-span, i.e. the period between the starting time of the first YC operation and the finishing time of the last YC operation. Lee *et al.* (2006) considered 2 YCs working in 2 non-overlapping zones with a SA algorithm to minimize the make-span. Cao *et al*. (2008) considered Double-Rail-Mounted gantry (DRMG) crane systems where two YCs can pass through each other along a row of blocks with a combined greedy and SA algorithm to minimize the loading time of containers.

In many works presented in the past, the objective of the YC dispatching algorithm is to minimize the total (average) vehicle waiting time or to minimize the makespan, i.e. the total time taken to finish a set of jobs by the YC. While minimizing vehicle waiting times or minimizing makespan often help reduce the vessel turnaround time, they may result in vehicles getting to the quayside earlier than they are needed. A more effective way to help reduce vessel turnaround time is to help minimize QC waiting time for a vehicle. This translates to minimizing vehicle job tardiness at a yard block. The vehicle job tardiness is with respect to the time a vehicle is scheduled to finish the transfer of container from/to a yard block. This time is referred to as the deadline of the vehicle job. It is not with respect to the time the vehicle arrives at the yard block.

For a loading job, based on *the time a QC needs the vehicle at the quayside*, the time this vehicle should leave the yard block with the container to travel to the quayside (the deadline of this vehicle job) can be derived assuming no traffic congestions. The deadline of the job is the time the QC needs the vehicle minus the expected travel time from the yard block to the QC. Depending on which vehicle is assigned this loading job, *the vehicle's arrival time at the yard block* can be derived or predicted. Given the deadlines of the vehicle jobs and their predicted arrival times at the yard block, the YC dispatching algorithm computes its serving sequence to minimize the average (total) job tardiness, which in turn minimizes QC waiting time for these vehicles.

For an unloading job, even though less critical to the QCs, longer times needed to store a container in the storage area also directly lead to interruptions of QCs' unloading process (Kemme, 2010). A deadline set for the vehicle will help it finish the current job and return to the QC on time to get the next container. Even though it does not matter which vehicle returns to the QC first, it is still important to have a stream of vehicles arriving at the QC continuously but with some intervals in between. In this way when a vessel is unloaded, quayside will not be crowded with early arriving vehicles but the QC does not need to wait for vehicles either.

For external vehicles carrying export or import containers, a deadline for a storing or retrieving job at the yard block will allow the terminal operator to guarantee a quality of service to the external truck companies.

We propose two provably optimal algorithms for an YC to handle the jobs in its assigned zone within a planning window efficiently. The algorithms take the predicted job arrival times and deadlines as inputs and minimize average job tardiness. We evaluate our algorithms against the Earliest Due Date first (EDD) and the Smallest Completion time Job First (SCJF). Our algorithms could find the best dispatching sequence with reasonable computational time in solving problems of practical sizes.

The rest of the paper is structured as follows. Firstly, a formal description of the YC dispatching problem is given in Section 2. Then two new algorithms are proposed in Section 3 and the experimental evaluations of the proposed algorithms are presented in Section 4. Conclusion is drawn in the last section.

2 PROBLEM FORMULATION

2.1 General Description

The following assumptions are made in the YC dispatching model:

- Each vehicle job involves one container only. If a vehicle carries two containers which are to be stacked at different locations, it will go to the delivery location of the top container first. A vehicle carrying multiple non-stacking containers to the same destination slot location is modeled as multiple container jobs as explained later.
- In each working zone in the yard, there is only one YC. Vehicles come to the zone for loading or unloading of containers.
- The deadlines of vehicle jobs and the slot locations (yard bays) of jobs are pre-determined.
- The vehicle arrival times can be predicted for a relatively short planning window, e.g. 30 minutes, and are given.
- YC gantry time between two job positions could be predicted with high accuracy as gantry speed is usually quite consistent.

In our formulation, the following notations are used:

$J = \{1, 2, \ldots, n\}$,	the set of job IDs in a YC's working zone for a planning window
a_i	the arrival time of job i.
p_i	the process time of job i by an YC.
d_i	the deadline of job i
m_{ij}	the time for YC gantry from the position of job i to that of job j.
S_i	the time an YC starts processing job i.
C_i	the time an YC completes processing job i.

J is the set of jobs to be sequenced. m_{0j} is the YC gantry time from its position at the start of the time window to the position of job j. C_0 is the time the YC is available to start to move to the position of its first job in the YC dispatching sequence.

For a set J of n vehicle jobs with predicted job arrival times a_i (i = 1, 2, …, n), deadline d_i (i = 1, 2, …, n) and YC gantry times m_{ij} (i = 0, 2, …, n; j = 1, 2, …, n), the tardiness of a job J_i is defined as $T_i = max\ (0, C_i - d_i\ )$, where C_i is the completion time of J_i. The objective of YC dispatching is to find a sequence so as to

$$\text{Minimize } \frac{1}{n}\sum_{i\in J} T_i$$

The completion time for job i is equal to its start time + process time, that is, $C_i=S_i+p_i$. When vehicle arrivals cannot be predicted, an YC can only start to move towards the next job location after the actual job arrival. Job starting time in this case could be derived as in Equation (1) where job i is the current job and job j is the next job. If vehicle arrivals can be predicted and the next job is decided, an YC is able to start moving towards the next job location before the actual vehicle arrival. This is referred to as the pre-gantry ability. Job starting time with pre-gantry ability is shown in Equation (2). The advantage of the pre-gantry ability is the possibility of utilizing YC idle time between jobs to transfer between different job locations.

$$S_j = \max(C_i, a_j) + m_{ij} \tag{1}$$

$$S_j = \max(C_i + m_{ij}, a_j) \tag{2}$$

The YC dispatching model is flexible to include operation conditions where some vehicles carry more than one container not stacked on top of each other, to be loaded/unloaded at the same slot location. They could be simply modeled as several container jobs with the same arrival time but different deadlines. For situations where several containers are to be loaded/unloaded at the same slot locations one after another, they will be individual jobs with their own vehicle arrival times, possibly one after another. In both situations, the YC dispatching algorithm will find a job serving sequence which returns the minimum total tardiness for all the jobs. The average job tardiness is the total tardiness divided by the number of jobs.

2.2 YC Dispatching Reduced to a Tree-search Problem

We believe that solving the YC dispatching problem as described in the last section by integer programming approach will not be feasible for practical applications. Given an YC dispatching problem of n jobs, there are n! possible dispatching solutions in total. The solution space can be transformed into a tree-search problem as shown in Figure 1. The root of the tree is the start node before the first job is selected. Each path from the start node to a leaf node in the tree represents a complete dispatching sequence of height n. The edge weight from node i to node j has a value equal to the tardiness of job j if the YC is to do job j immediately after finishing job i. This edge weight is given by

$$W_{ij} = \max \{0, \max (C_i + m_{ij}, a_j) + p_j - d_j\} \tag{3}$$

Note that m_{0j} is the YC gantry time from its position at the start of the job sequence to the position of job j and C_0 is the time the YC is available to start to move to the position of its first job in the YC dispatching sequence. The task is to find a path of minimum total distance (i.e. minimum total tardiness for a given n) from the start node to a leaf node that visits each job exactly once. From (3), it can be seen that each edge weight is not pre-defined and depend on i. In addition, the YC process time p_j depends on not only i but also all other predecessor jobs in the path from s to j. The YC process time depends on whether this job is at the top of the stack when YC comes to serve this job. A job which is not at the top of the stack at the beginning of the job sequence may become top of the stack if the containers above it are loaded onto vehicles in the early part of the job sequence (we assume that containers at the top of a stack at the beginning of the job sequence will not be blocked by other containers during reshuffling in the early part of the job sequence). p_j needs to be dynamically computed in the process of searching for the optimal job sequence.

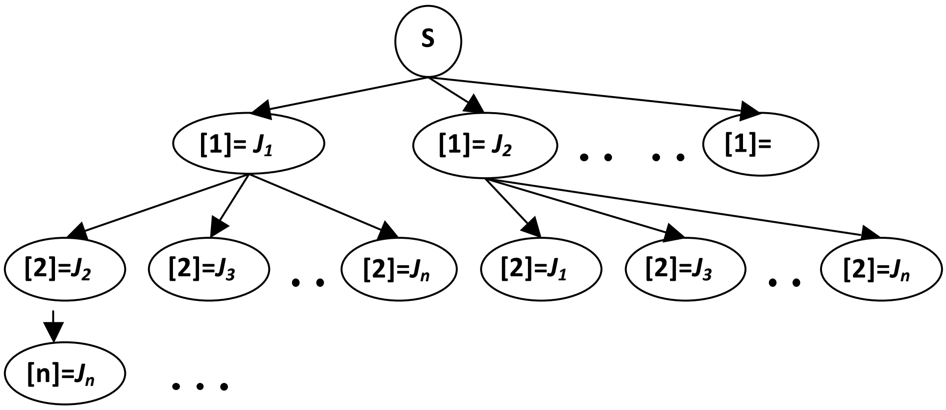


Figure 1. Search Space of the Problem

3 OPTIMAL SEARCH ALGORITHMS

In finding the least-cost path of the YC dispatching problem, we need to traverse the tree to search for the optimal solution. As the problem is strongly NP-hard, exhaustive search would be time-consuming to perform. Here we propose to use modified A* search to reduce search time and yet to guarantee optimality. We derive a heuristic function using domain knowledge of YC operations for the modified A* search.

3.1 Modified A* Search: MTA*

A* search is a common method for graph search to find the optimal path from an initial node to one goal node. It keeps an *open list* of nodes to be expanded and always tries to select the most promising node first based on an evaluation function $f(x)= g(x) + h(x)$ where $g(x)$ is the cost from the start node to x and $h(x)$ is the estimated lowest cost from x to a goal node. If h(x) never overestimates the true cost h*(x), i.e. h(x)≤h*(x), the heuristic h(x) is admissible. When coupled with an admissible heuristic $h(x)$, A* search is optimally effective.

Unlike the original A* search, we are not actually looking for one “goal” node in the YC dispatching problem. The search tree is formed such that each path from the root to a leaf node is of height n representing a complete dispatching sequence of n jobs. The objective of our tree search is to find a path from the start node to a leaf node (a dispatching sequence of n jobs) with minimum f(x) (i.e. total job tardiness). Once a leaf node is reached, the cost f(x) of this complete dispatching sequence is kept.

We still use the A* cost function $f(x)$ to select the most promising node to expand, but the criterion for stopping the search is modified accordingly. In the original A* search, the search will stop when a goal node is reached or when the open list is empty. In the modified A* search, the search will stop when we reach a leaf node with a cost f(x) lower than that of any path in the open list or when the open list is empty. In this way, we formed our definition of the “goal state” in the stopping criteria of A* search while maintaining the characteristics of A*search: admissible, complete and optimally effective. Pseudocode of this Modified A* algorithm MTA* is shown in Figure 2. For this algorithm, we have

$g(x)$ = cumulated tardiness from start to node x with edge weight W_{ij} by Equation (3).
The way to compute job processing time dynamically is presented in Section 3.3.
$h(x)$ = estimated lower bound cost from node x to a leaf node.
N = set of all jobs, $|N| = n$
$P(x)$ = set of jobs already in the (partial) path from the root of the tree (start) to node x
$U(x) = N – P(x)$

A* is optimal in tree-search if *h(x)* is an admissible heuristic, that is, *h(x)* never overestimates the cost from *node_cur* to the goal. In this problem, an admissible heuristic means *h(x)* never overestimates the total job tardiness to handle the remaining unordered jobs.

```
MTA*  // the cost function f(x) determines the priority of node x
 Create a node containing start status as node_start
 Create a node containing infinite cost as node_best
 Put node_start in the priority queue q_open
 WHILE q_open is not empty
    node_cur = remove_first (q_open)
    IF f (node_cur) is no better than f (node_best)
        Return node_best
    IF node_cur is a leaf node
        Update node_best as node_cur
    ELSE
        Generate each node_successor of node_cur
        FOR each node_successor x
                Get true cost g(x) from start to this node
                Estimate lower bound cost h(x) from this node
                Get cost function f(x) = g(x) + h(x)
                Add x node to priority queue q_open according to f(x) value
```

Figure 2. Pseudocode of Modified A* Search

Now, the problem is how to evaluate *h(x)* where *x* is a successor node of *node_cur* in the algorithm presented in Figure 2. In our notation, *U*(*node_cur*) is the set of jobs not included in the path of the tree from the root to *node_cur*. Let *M*= |*U(node_cur)*|.

Consider a node $x = J_i$ where $J_i \in U(node\_cur)$. To evaluate *h(x)*, the cost from J_i to a leaf node, we need to estimate the minimal total job tardiness for the remaining M-1 jobs, that is, $U(x) = U(node\_cur) - \{J_i\}$. *U(x)* can be partitioned into two groups. One group F_{later} is the set of jobs that will definitely miss their deadlines if J_i is the next job after *node_cur*. These are the jobs such that even if they are processed immediately after the completion of J_i, their completion time will be after their deadlines. The other group $F_{earlier}$ is the rest of the jobs. In other words, a job $j \in U(x = J_i)$ is in F_{later} if its completion time would be after d_j even when it is the immediate next job to be served after job J_i. We define

$$U(x = J_i) = F_{earlier} \cup F_{later}$$

$$F_{later} = \{J_j\},\ where\ C_i + m_{ij} + p_j > d_j$$

$$F_{earlier} = \{J_j\},otherwise$$

For each job in F_{later}, the minimum tardiness is $\max(C_i + m_{ij}, a_j) + p_j - d_j$, because the earliest time job *j* may be started after *i* is $\max(C_i + m_{ij}, a_j)$. The minimum total tardiness for jobs in F_{later} is shown in Expression (4).

$$\sum \left(\max(C_i + m_{ij}, a_j) + p_j - d_j\right),\quad j \in F_{later} \tag{4}$$

In order never to overestimate the total tardiness in Equation (4), we assume the minimum value for p_j. So p_j will assume the value where the job has no need for the additional reshuffling of the containers above it. In other words, we assume that containers to be retrieved from the yard during loading are on top of the storage yard, and containers to be stored in the yard during unloading are just placed on top of the proper slot. In this case, process time p_j is the time YC makes one container move, T_p.

In addition to the tardiness described in Expression (4), the second job to be handled in F_{later} ends in additional tardiness of at least one T_p, the minimum processing time of the first job handled in F_{later}. The third job to be handled in F_{later} ends in additional tardiness of at least $2*T_p$, the sum of minimum job process times of the first and the second job handled in F_{later}. Likewise, the last job to be handled in F_{later} will end in additional tardiness of at least $(|F_{later}|-1)T_p$, the sum of the job process times for the previous $(|F_{later}|-1)$ jobs in F_{later}.

Thus, combining Expression (4) and the tardiness mentioned above, the minimum total tardiness for jobs in F_{later} is as follows.

$$[\sum(\max(C_i+m_{ij},a_j)+T_p-d_j)]+[1+2+...+(|F_{later}|-1)]*T_p\ ,\ \ j\in F_{later} \tag{5}$$

Jobs in $F_{earlier}$ are any job j for which $C_i+m_{ij}+T_p\le d_j$. However, tardiness for a job in this set will happen if a_j is after C_i+m_{ij} and its completion $a_j+T_p>d_j$. The minimum total tardiness for these jobs is described in Expression (6)

$$\sum Max\ \{0,\max(C_i+m_{ik},a_k)+T_p-d_k\},\ \ k\in F_{earlier} \tag{6}$$

Estimated minimum cost from node J_i to a leaf node $h(x=J_i)$ is now labeled as $h(i)$. The heuristic $h(i)$, which describes the minimum total job tardiness for the remaining unplanned jobs, is then the sum of the two components in (5) and (6):

$$[\sum_{j\in F_{later}}(\max(C_i+m_{ij},a_j)+T_p-d_j)]+[1+2+...+(|F_{later}|-1)]*T_p+$$

$$\sum_{k\in F_{earlier}}Max\ \{0,\max(C_i+m_{ik},a_k)+T_p-d_k\} \tag{7}$$

Proof: *h(i)* is admissible

(1) The tardiness as expressed by (5) is the very minimum for jobs in F_{later} because (i) each job has to at least wait until Job J_i is complete and move to its location; (ii) it is the total tardiness for these jobs when no job in $F_{earlier}$ would be served before any job in F_{later} and the YC gantry time between jobs is not considered except the first job after J_i.

(2) The tardiness as expressed by (6) is the very minimum for each of the jobs in $F_{earlier}$ as if it were the first job to be served after job J_i.

Components (5) and (6) do not overlap with each other, h(i) can never overestimate the total tardiness for the remaining M-1 jobs. Thus h(i) can never overestimate the cost to the leaf node and is hence admissible.

3.2 Prioritized Recursive Backtracking with Heuristics: MT-RBA*

A* algorithm is optimally efficient for any given heuristic function. However it has two limitations in real world applications. First one is memory usage. Because A* keeps all generated nodes in memory and the number of nodes is increasing exponentially as the problem size increases, it may runs out of memory space. This limits its usage in large-scale problems. The second limitation is that real time dispatching often requires anytime algorithms. This means that an algorithm could work out a solution very quickly and keeps improving it until the optimal is reached or interrupted by real time constraints to provide the current "best" solution. A* search has actually a best-first search feature and thus may take a long time to obtain a solution (i.e., not anytime) in the worst case or near-worst case.

To overcoming these limitations, we propose a Recursive Backtracking (RB) based algorithm. RB is complete and optimal if the depth of a search tree is finite and there is no time constraint. It greatly reduces memory usage by recursively calling sub problems and keeping only the nodes on current path. RB could quickly find the first solution by visiting only N nodes in a problem of size N. It would then backtrack to examine other solutions. If a better solution is encountered during the backtracking process, the knowledge of current best solution will be updated accordingly. Therefore, RB based algorithm is anytime such that a current best solution is provided even though the entire search is not finished. RB is able to overcome the problem of memory overflow. However, it needs to traverse the entire tree to find the optimal solution, which is time-consuming.

We propose to employ heuristics to trim the search space. Every time we expand a new node along a search path, the job tardiness could be obtained and the total job tardiness of all planned jobs from the start to the current node g(x) is updated. The pruning of the search space could be more effective if the lowest cost from the current node *x* to a goal node can be computed. Let h(x) represents the estimated minimum total tardiness for all unplanned jobs. If h(x) is admissible, in other words, h(x) never overestimates the cost from the current node to a goal node, the search is by nature optimal because it thinks the cost of solving the problem is less than it actually is. Since g(x) is the exact cost from start node to the current node x, this implies that g(x)+h(x) never overestimates the true cost of a solution through node x. The decision to stop searching further the sub-trees of the current node could use the evaluation of g(x)+h(x). This evaluation function has been proposed in last section. By combining this heuristic into the RB algorithm, we propose the MT-RBA* which will not miss the optimal solution and will greatly reduce the computation time of RB. Pseudocode is shown in Figure 3.

```
J = {J1, J2, … , Jn}
YCDispatching (J) {
     newJ = ϕ; optimalJ = ϕ;CurSmallest = ∞;
     MT-RBA*(J, newJ);
}
MT-RBA*(J, newJ) {
    FOR each job Ji ∈ J    // Select Ji as the job to serve after jobs in newJ;
          Remove  Ji from J and Append  Ji  to newJ;
          Get g( Ji ) for jobs in newJ by (3)
          Estimate lower bound cost h( Ji ) for jobs in J by (7) ;
      IF g( Ji )+h( Ji ) is smaller than CurSmallest
          IF J is not empty
                 MT-RBA*(J, newJ);
      ELSE  // TotalWaitingT is smaller than CurSmallest
                 Update CurSmallest as  g( Ji )+h( Ji ) ;
                 Update optimalJ = newJ;  //Store Optimal List
 }
```

Figure 3. Pseudocode of MT-RBA*

In Figure 3, optimalJ is created to record the current best complete dispatching sequence. In each iteration of the FOR loop in MT-RBA*, an unplanned job J_i will be considered as the current job to be served by the YC. The tardiness $g(J_i)$ from the start node till job J_i is the partially built dispatching sequence stored in newJ. The minimum total tardiness for unplanned jobs $h(J_i)$ is computed according to Equation (7). The sum $g(J_i)+h(J_i)$ represents the estimated minimal total tardiness for any dispatching sequence which has the partially built dispatching sequence as in newJ. If this sum is smaller than that of the current best dispatching sequence, the function MT-RBA* will be called to explore further the current path in the search tree. Otherwise, the current path is terminated and another unplanned job will be considered as the current job. If the current job is a leaf node, the algorithm has found a complete dispatching sequence and the current best solution will be updated accordingly.

In each iteration, after picking job J_i as the current job, it results in a partial built dispatching sequence $P(J_i)$ = newJ, and jobs left unplanned $U(J_i)$ = J. Therefore there are *factorial* ($|U(J_i)|$) dispatching sequences with the partially built dispatching sequence newJ as the prefix. Whenever $g(J_i) + h(J_i)$ is found to be no better than the current best solution, a total of *factorial*($|U(J_i)|$) dispatching sequences are pruned from the search space. However, in the worst case, the optimal solution could be reached last and we could still be forced to explore the entire solution space. Therefore, discovery of a good path which has near-optimal total tardiness at the early stage of the tree search is crucial to the performance of MT-RBA*.

As shown in Figure 3, MT- RBA* takes a list J of unplanned jobs as input. For each job J_i in list J, it will be considered as a next job and followed by a function call for the remaining jobs $(J- \{J_i\})$.To accelerate the search process, we proposed a technique called prioritized search order which is more likely to discover a good dispatching sequence early in the planning process, instead of choosing the next job randomly. The prioritized search order we use is the ascending order of the job deadlines. Intuitively, if the YC serves a job with an earlier deadline first, it will contribute to the minimization of total job tardiness.

3.3 Dynamic computation of YC service times $\mathbf{p_j}$

In general, the time taken by a YC to load/unload a container ($\mathbf{p_j}$ in Equation (3)) does not vary a lot provided that the container is at the top of the stack. However, there are loading/unloading jobs that involve containers not at the top. In such situations, the YC has to do reshuffling to remove the containers above the wanted container (each is roughly the time for one container move). Whether a wanted container is at the top or not is sequence dependent so cannot be pre-determined in the search for the optimal job sequence. Figure 4 gives the algorithm to manage the reshuffling and compute each job processing time dynamically according to the current job sequence (Equation (3) used in computing g(x) in MTA* and MT-RBA*. The yard block status is maintained separately for each job sequence. When a job is found not to be on top, the containers above this container will be moved to the nearest row in the same slot (bay) without blocking a future job. The job processing time for each job is a multiple of the number of containers moved, including the wanted container.

```
IF container at top tier // check the yard block status
    Set job process time pj to TimeForOneMove, Tp
ELSE // check how many container above  {
    NumToReshuffle = current stack height – tier number of the job
    FOR each container to be reshuffled {
     Move the container to the nearest row in the same slot(bay)
            without blocking an unplanned job;
     update the location of the moved container
    }
    Set job process time pj to (NumToReshuffle + 1) * Tp
}
```

Figure 4. Dynamic computation of job processing time $\mathbf{p_j}$.

4 PERFORMANCE EVALUATION

4.1 Design of experiments

To evaluate the performance of the proposed YC dispatching algorithms, simulation experiments were carried out. The simulation model is programmed using C++ language under Visual C++ 6.0 compiler on Pentium Core2 Quad CPU Q9450 and 3GB RAM. Parameter settings in the experiments were obtained from real world terminal models as in past projects (Guo *et al*. 2007). The linear gantry speed of an YC is 7.8km/hour. A yard block has a size of 36 slots. We conducted experiments where a YC is in charge of one yard block. A planning window of 10 jobs for a YC is simulated.

Vehicle jobs arrive at the yard block at four different arrival rates. We assume that the vehicle inter-arrival times follow exponential distributions. Mean inter-arrival time is set to 180, 240, 300 and 360 seconds respectively. The slot locations of the jobs are generated randomly within the zone. Other recent studies using randomized container locations include, for example, Zeng and Yang (2009).

The deadlines depend on the type of jobs: relatively tight deadlines for loading jobs and less tight deadlines for unloading jobs. This reflects the fact that it is more important to finish loading jobs on time than unloading jobs. Following the design of Chu (1992), deadlines are generated from a uniform distribution. Let T = a vehicle's arrival time + 2 minutes. 2 minutes is the YC time to load/unload one container from/to the top of a stack, not including the gantry time. For a vessel loading job, its deadline is a random variable from a uniform distribution with a width of 360 seconds around T such that 95% of the time its deadline is after T (that is, a uniform distribution ranging from 18 seconds before T and 342 seconds after T). Since vehicle arrivals at the yard side may be late in special cases like traffic jams or machine break downs, 5% jobs will have deadlines earlier than T. For a vessel unloading job, the uniform distribution has a range of 720 seconds around T. 5% jobs will have deadlines earlier than T. For external vehicle jobs, the deadline is 30 minutes after the truck's arrival.

The types of jobs that arrive at the yard block are a mixture of vessel loading/unloading jobs and external truck loading/unloading jobs. We tested a mixture of jobs as shown in Table 1. The percentage of each type of jobs is set which is similar to the scenarios in a terminal with high volumes for transhipment containers. When the tier number of a job is not the top tier, reshuffling may be needed depending on the job sequence as discussed in Section 3.3.

The algorithms MTA* and MT-RBA* are evaluated against the greedy heuristics Earliest Due Date first (EDD) and Smallest Completion Time Job First (SCJF). EDD and SCJF could find a solution fast but the optimal solution is not guaranteed. EDD generates job sequences in increasing order of jobs' deadlines. SCJF is a heuristic algorithm by Ng (2005). For each experimental setting, 30 independent runs were performed.

Table 1. Mixture of jobs.

Percentage	Job type	Tier number of job	Percentage
40%	Vessel loading	Top tier	50%
		2nd top tier	30%
		3rd top tier	20%
40%	Vessel unloading	Top tier	100%
10%	External truck loading	Tier 1, 2, 3, 4	Equal probability
10%	External truck unloading	Top tier	100%

4.2 Results and discussions

Table 2 shows the average job tardiness of the four algorithms tested for the four average inter-arrival time scenarios. Since MTA* and MT-RBA* compute the optimal job sequence for minimizing job tardiness, their results are the same even though the resulting job sequences may be different. As the average job inter-arrival time increases, average job tardiness decreases. However, the results confirm that our optimization algorithms make significant improvements from the greedy heuristics in all tested scenarios.

Table 2. Average job tardiness (seconds).

	IAT 180	**IAT 240**	**IAT 300**	**IAT 360**
MTA*	32.15	26.75	23.08	19.67
MT-RBA*	32.15	26.75	23.08	19.67
EDD	59.07	47.55	44.25	38.41
SCJF	74.96	58.04	47.94	36.90

Table 3. Computational time (seconds).

	IAT 180	**IAT 240**	**IAT 300**	**IAT 360**
MTA*	437.2	256.0	153.0	131.6
MT-RBA*	28.4	2.8	2.0	2.3
EDD	0.1	0.1	0.1	0.1
SCJF	0.1	0.1	0.1	0.1

Table 3 shows the average computational times taken by the four algorithms tested for the four average inter-arrival time scenarios. As expected, the greedy heuristics EDD and SCJF takes little computational time, much faster than MTA* and MT-RBA*. MTA* is more time consuming than MT-RBA*. Although MT-RBA* explores more nodes in the search tree than MTA* method because MTA* is optimally effective, it outperforms the MTA* method because it does not have the overhead to maintain the open list queue for deciding which node to visit next. The prioritized search order in MT-RBA* also helps to get a good solution at the beginning of the search which effectively prunes many branches of the search tree at early stage, reducing computational time.

For both MTA* and MT-RBA*, the computation for heavy workload scenario (average inter-arrival time of 180 seconds) is much higher than the other scenario. With our mixture of the different job types as shown in Table 1, the average YC service time is 166 seconds without including inter job gantry time. So this represents the situation where the YC is almost 100% busy. In such a situation, jobs have smaller inter-arrival time differences and the optimal sequence is harder to find. This can be explained using queuing theory: the closer the job arrival rate is to the process rate (YC handling rate), the higher probability there is to see queuing jobs. When several vehicle jobs wait for the YC service, their relative job locations would be an important factor in determining the optimal service order. The algorithms may need to expand and evaluate more alternative job sequences in finding the optimal, resulting in longer computational times. However, MT-RBA* takes less than half a minute to find the optimal solution.

5 CONCLUSION

We propose to compute job sequences for YCs to minimize vehicle job tardiness instead of minimizing vehicle waiting time. This helps vehicles to reach the QCs with minimal delay and therefore reduce vessel turnaround time. We present a modified A* search algorithm MTA* with an admissible heuristic to compute optimal YC dispatching solutions to minimize vehicle job tardiness in the storage yard. To overcome the large memory usage limitation of the MTA* search, we further present MT-RBA* algorithm that combine the advantages of A*search and Backtracking with prioritized search order. Experiments were carried out to evaluate the algorithms under four levels of job arrival rates. Our results show that the proposed algorithms consistently perform very well over all tested cases, significantly reducing the vehicle job tardiness.

Future work includes more comprehensive evaluation of our algorithms. For example, a different distribution of the job deadlines may be used. The robustness of the algorithms can be evaluated by incorporating uncertainties in vehicle job arrivals.

ACKNOWLEDGEMENTS

This project is supported by the NOL Fellowship program.

REFERENCES

Cao, Z. Lee, D. H. and Meng, Q. (2008) Deployment strategies of double-rail-mounted gantry crane systems for loading outbound containers in container terminals, *International Journal of Production Economics*, 115, 221–228.

Guo, X. Huang, S. Y. Hsu, W. J. Low, M. Y. H. Chan, T. H. and Liu, J.H. (2007), Vehicle Dispatching with real time location information in container terminals, In *Proceedings of the European Modeling and Simulation Symposium.*

Guo, X. Huang, S. Y. Hsu, W. J. and Low, M. Y. H. (2011), Dynamic Yard Crane Dispatching in Container Terminals with Predicted Vehicle Arrival Information, *Advanced Engineering Informatics*, 25(3), 472-484.

Jung, S. H. and Kim, K. H. (2006), Load scheduling for multiple quay cranes in port container terminals, *Journal of Intelligent Manufacturing*, 17, 479–492.

Kemme, N. (2010), RMG crane scheduling and stacking, In J. W. Bose (eds), *Handbook of Terminal Planning*, Springer.

Kim, K. H. Kang, J. S. and Ryu, K. R. (2004), A beam search algorithm for the load sequencing of outbound containers in port container terminals, *OR Spectrum*, 26, 93–116.

Kim, K. M. and Kim, K. Y. (1999), An optimal routing algorithm for a transfer crane in port container terminals, *Transportation Science,* 33(1), 17–33.

Kim, K. Y. and Kim, K. H. (2003), Heuristic algorithms for routing yard-side equipment for minimizing loading times in container terminals, *Naval Research Logistics*, 50, 498–514.

Lee, L. H. Chew, E. P. Tan, K. C. and Han, Y.B. (2006), An optimization model for storage yard management in transshipment hubs, *OR Spectrum*, 28, 539-561.

Li, W. Wu, Y. Petering, M. Goh, M. and d. Souza, R. (2009), Discrete time model and algorithms for container yard crane scheduling, *European Journal of Operational Research*, 198, 165–172.

Ng, W. C. (2005), Crane scheduling in container yards with intercrane interference, *European Journal of Operational Research*, 164, 64–78.

Ng, W. C. and Mak, K. L. (2005), An effective heuristic for scheduling a yard crane to handle jobs with different ready times, *Engineering Optimization*, 37(8), 867-877.

Steenken, D. Voβ, S. and Stahlbock, R. (2004), Container terminal operation and operations research – a classification and literature review, *OR Spectrum*, 26, 3-49.

Zeng, Q. and Yang, Z. (2009), Integrating simulation and optimization to schedule loading operations in container terminals, *Computers & Operations Research*, 36(6), 1935–1944.

STORAGE CHARGING AND RE-MARSHALING FOR OUTBOUND CONTAINERS IN CONTAINER TERMINALS

Kap Hwan Kim

Department of Industrial Engineering, Pusan National University, Jangjeon-dong, Gumjeong-gu, Busan 609-735, Korea, Tel: +82-51-510-2419, kapkim@pusan.ac.kr

ABSTRACT

This paper discusses a method of determining the storage charge and planning the re-marshaling operation for outbound containers in a container yard. The space is pre-allocated to outbound containers of each vessel and a layout plan for the outbound containers is prepared in advance before they arrive at the terminal. Because the space that can be allocated to each vessel is limited, a storage fee is charged for restricting the duration-of-stay of containers at the yard and the re-marshaling operation is performed for efficiently using the storage yard. This study analyzes the impact that these operational tools have on the space requirement, the cost, and the revenue for terminal operators. Some mathematical formulations are suggested to optimize the decision parameters for the storage charge, including the free-time-limit and the storage charge per day, and to determine the re-marshaling time.

Keywords: Container terminal, pricing, re-marshaling, outbound containers

1 INTRODUCTION

Storage space is an important resource in container terminals. Efficient utilization of the storage space significantly improves the productivity and the profitability of the container terminal. In container terminals with parallel layouts, relative positions between containers, which are retrieved successively, should be close to each other so that the travel distance of the yard crane can be minimized. For the purpose, containers of the same size bound for the same destination port, which is called containers of the same group, are stored in the bays close to each other. That is, several bays are allocated to containers of the same group. Usually bays in several blocks are allocated to a vessel. The layout plan for the storage of outbound containers is prepared in advance. Figure 1 illustrates a layout plan for outbound containers. When outbound containers arrive at the yard, they are stored at the locations according to the layout of the storage. Containers arrive at the yard before the layout plan is prepared are stored at a temporary storage area in which they are stacked in a random manner. When the storage space becomes available and the layout is constructed for the permanent storage, the outbound containers in the temporary storage area are moved to the permanent storage area according to the layout, which is called "re-marshaling operation."

The space requirement for the outbound containers of a vessel depends on the distribution of the duration-of-stays (DOSs) of containers at the yard of a container terminal before the cutoff time (DOSY) and the time when the re-marshaling operation is done. To encourage customers not to store containers for a long time, terminal operators usually charge a storage fee - which is higher than the storage fee in an outside container depot - for storing a container beyond a pre-specified free-of-charge time limit (free-time-limit). The structure of the storage charge affects the distribution of DOSYs significantly.

Bay 1	Bay 2	Bay 3	Bay 4	Bay 5	Bay 6	Bay 7	Bay 8	Bay 9	Bay 10
SIN	SIN	SIN	SIN	RTM	RTM	RTM	HAM	HAM	SIN
SIN	SIN	SIN	SIN	RTM	RTM	RTM	HAM	HAM	SIN
SIN	SIN	SIN	SIN	RTM	RTM	RTM	HAM	HAM	SIN
SIN	SIN	SIN	RTM	RTM	RTM	HAM	HAM	HAM	SIN
SIN	SIN	SIN	RTM	RTM	RTM	HAM	HAM	HAM	SIN
SIN	SIN	SIN	RTM	RTM	RTM	HAM	HAM	HAM	SIN

Figure 1. An illustration of a layout plan for outbound containers of various container groups (top view) (Woo and Kim, 2011)

This paper addresses the storage charging and re-marshaling problem for storage of outbound containers. There is a tendency that most containers are delivered within several days before the cutoff time which is the deadline for outbound containers to enter a terminal gate before the corresponding vessel arrives at the terminal. Watanabe (2001) suggested the exponential distribution as the distribution for duration-of-stays at the yard of a terminal (DOSYs).

There are two possible storage and delivery schedules for outbound containers. Outbound containers may be delivered from a shipper directly to the container yard (CY) in the port container terminal (PCT). However, the storage charge must be paid by the shipper to the terminal operator when the container is delivered earlier than the beginning of the free-time-limit. In the other way, outbound containers may be moved to an outside container depot, which is operated by a private company and called an off-dock container yard (ODCY) and charges a lower storage fee than in the PCT, stacked for a period of time and then moved again to the CY in the PCT at the beginning of the free-time-limit. When the terminal operator provides a free-time-limit, it is always better to utilize the benefit of the free-time-limit than to store containers at an ODCY at a storage cost. The delivery and storage schedule is selected based on the delivery time requested by its shipper and the costs for storing and transporting it from the shipper to the PCT.

Until recently, little attention has been paid by academic researchers to the re-marshaling operation and the pricing strategy for outbound container inventories in container terminals. Regarding the re-marshaling operation, Taleb-Ibrahimi *et al.* (1993) have approached the re-marshaling operation analytically for the first time in which temporary storage areas are provided for containers that arrive before a designated storage space has been allocated for them. The time to allocate designated space to each vessel is determined using a trade-off between the cost of the temporary storage space and the cost of relocating the containers from the temporary storage area to the designated space. Woo and Kim (2011) addressed the problem of estimating the space requirement for outbound containers in port container terminals.

The following three papers are directly related to the pricing issue in this paper. Castilho and Daganzo (1991) addressed a pricing problem for temporary storage facilities at ports. They analyzed customer behaviors for different price structures when they maximize their cost saving – resulting from using temporary storage space instead of warehouses at a far distance – minus the sum of moving expenses and holding costs. Holguin-Veras and Jara-Diaz (1999) proposed a method for determining space allocation and prices for priority systems in CYs. They analyzed three different pricing rules: welfare maximization, welfare maximization subject to a breakeven constraint, and profit maximization. For expressing the effect of the congestion in a CY on the handling cost, they assumed a continuous function whose variables are the storage amount and the capacity of a CY. They extended this study to the case with elastic arrivals and capacity constraint (Holguin-Veras and Jara-Diaz, 2006).

Kim and Kim (2007) proposed methods for determining prices for the storage of inbound containers in CYs. The study was based on detailed cost models of handling activities in CYs for customers and terminal operators. Lee and Yu (2012) proposed inbound container storage pricing game models between the container terminal and a remote container yard.

The main contribution of this paper is that this study proposes mathematical models, which are based on statistical models on deliveries of containers by shippers, for determining the storage charge and the re-marshaling time for outbound containers and proposes useful properties of the optimal solution. The next section analyzes the storage charging problem and section 3 discusses how to determine the re-marshaling time. The final section provides concluding remarks.

2 STORAGE CHARGING AND ITS IMPACT ON THE SPACE REQUIREMENT

This study assumes a storage charge that is proportional to the length of storage time beyond the free-time-limit. Thus, the price structure can be expressed by the free-time-limit (*F*) and the storage price per day (*S*) for the storage longer than the free-time-limit. This is a typical price schedule which is used in most container terminals in Korea. In Terminal A and B, F was 4 days, and S was approximately US$ 14.7 and US$ 18.9, respectively.

When a terminal operator proposes a price schedule of (*F, S*) to customers, customers will select a storage schedule for their containers based on costs and their duration-of-stays at the port before the cutoff time (DOSPs) which means the time between the arrival time of a container at the port and the cutoff time for the container. To store a container in an ODCY, an additional transportation cost between an ODCY and the PCT and the handling cost at the ODCY are incurred. The cost of a container can be represented by Figure 2-(a) as a function of DOSP. Note that the fixed cost for the ODCY results from the additional handling cost at the ODCY and the transportation cost between the ODCY and the PCT. Because the storage cost at the ODCY is lower than that at the PCT, containers whose DOSP are shorter than a specific time, which is denoted as T_s, will be moved to an ODCY first and stay at the ODCY until they are moved to the PCT at the beginning of the free-time-limit. Note that T_s is DOSP of a container at which the direct delivery to the PCT costs the same as storing it temporarily at an ODCY and then delivering it into the PCT just when the free-time-limit starts. Thus, from the viewpoint of the terminal operators, the curve for the revenue per container can be drawn as shown in Figure 2-(b).

A longer free-time-limit and a lower storage price at the PCT will induce customers to deliver their containers directly to the PCT. A shorter free-time-limit and a higher storage price result in more customers to move their containers to an ODCY before they are delivered to the PCT even at the additional transportation and handling costs.

This section attempts to suggest a method to determine the storage charge and the free-time-limit for maximizing the profit of the terminal operator and a method to determine the amount of space to be allocated to each vessel considering the distribution of duration-of-stays of containers at the yard of the PCT before the cutoff time (DOSYs) for a given storage charge and the free-time-limit.

Mathematical models in this paper are based on the following assumptions and notations.

Assumptions

(1) The storage charge at an ODCY is lower than that at the PCT. This is why containers are sometimes temporarily stored in an ODCY.

(2) The transportation cost between the PCT and an ODCY, the handling cost, and the storage charge are the same at all the ODCYs.

(3) When a space is provided for the permanent storage, the whole space required for storing all the outbound containers for the vessel is allocated at once. This is due to the

requirement that outbound containers should be stored according to the carefully planned layout as shown in Figure 1.

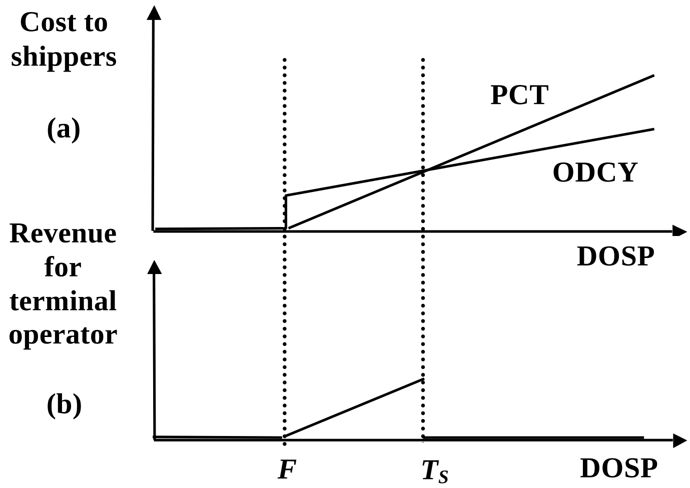


Figure 2. Cost function of a shipper and the revenue of a terminal operator

Notations

c_s = Storage space cost of a container at the PCT (US$/ TEU/ day)
s_o = Storage charge of a container at an ODCY (US$/ TEU/ day).
c_o = Transportation charge for a container (TEU) between the PCT and an ODCY plus the handling cost of a container at the ODCY (US$/ TEU).
u = Total amount of outbound containers to be loaded onto a vessel (TEUs)
a_{tot} = Total available space for outbound containers for all vessels (TEU-days)
X = DOSP of an outbound container. This is a continuous random variable.
f(X) = Probability density function (PDF) of X.
Y = DOSY of a container.
r(Y) = PDF of Y.
F = Free-time-limit for outbound containers (a decision variable).
S = Storage price in the CY of the PCT before the free-time-limit (F) (US$/ TEU/ day) (a decision variable).
A = Storage space allocated to outbound containers for a vessel (TEU×days) (a decision variable).

Suppose that Figure 3 shows the PDF of the DOSPs of outbound containers. When a price schedule is announced by the terminal operator, each customer schedules the storage and the transportation of his/her own container in a way of minimizing his/her total cost. As a result, considering their expected decisions, a new PDF of DOSYs, r(Y), can be derived (Figure 4). The PDF r(Y) can be evaluated as follows: all the containers for which temporary storage at an ODCY is more economical than directly delivering to the CY of the PCT will be moved to an ODCY. They will be moved again to the CY of the PCT at the beginning of the free-time-limit. All the containers which shippers want to deliver before a specific time, which we denote T_s in this study, are classified into this class. The other containers will be delivered to the terminal at the same time as in case that no storage fee is charged. Thus, r(Y) becomes

$$r(Y) = \begin{cases} f(Y) & \text{for } 0 \le Y < F \\ \int_{T_s}^{\infty} f(Y)dy & \text{for } Y = F \\ f(Y) & \text{for } F < Y \le T_s \\ 0 & \text{for } Y > T_s \end{cases} \qquad (1)$$

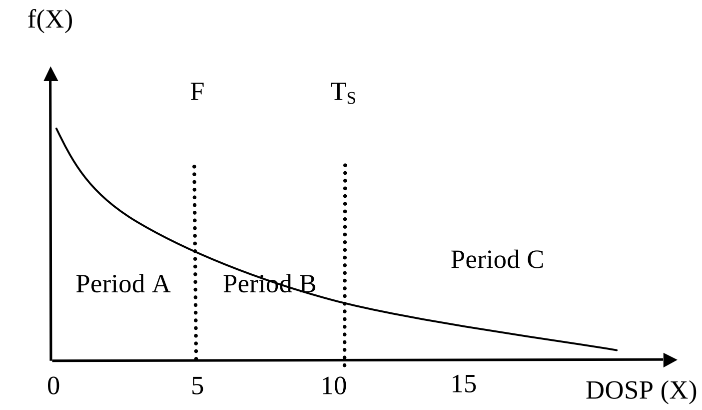


Figure 3. PDF of DOSPs

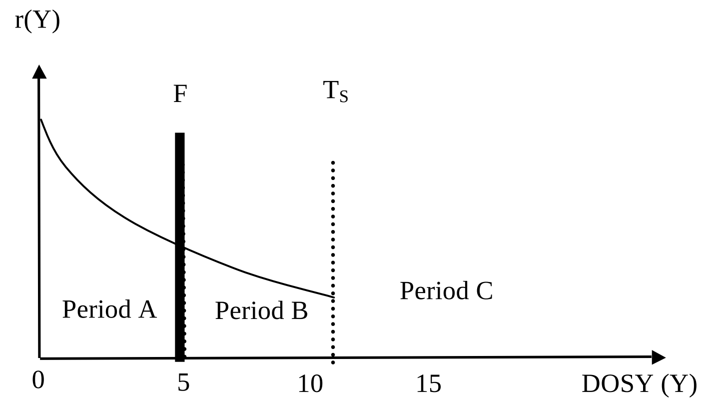


Figure 4. PDF of DOSYs of containers for a storage charge and a free-time-limit

From the point of view of terminal operators, they will attempt to maximize the revenue from the storage charge from all shippers. Thus, the problem of determining the price schedule for storing outbound containers can be formulated as follows:

(P1)

$$\max_{F,S,T_s} E(PF(F,S)) = \int_F^{T_s} S(x-F)f(x)dx \qquad (2)$$

subject to

$$S(T_s\text{-}F) = c_0 + s_0(T_s\text{-}F) \qquad (3)$$

$$S, F, T_s \geq 0.$$

Note that in (3), T_s is defined to be the DOSP of a container for which a direct delivery of the container to the PCT costs the same as a temporary storage at an ODCY and a delivery to the PCT at the beginning of the free-time-limit.

Property 1: For (P1), F=0 is an optimal solution.

Proof) Consider a price schedule (F_1, S_1) where $F_1 > 0$. And, consider another schedule (0, S_2) such that $T_{S1} = T_{S2} = t_0$. That is, the following holds by the definition of S_2: $c_0/(S_1\text{-}s_0) + F_1 = c_0/(S_2 - s_0)$. From this equality, it follows $S_2 < S_1$. Let the revenue from a customer with DOSP, *X*, under the price schedule, (*F*, *S*), be denoted by R(*X*|*F*,*S*). Then, because T_{S1} (= t_0) is defined to be the DOSP of a container for which the storage charge at the PCT equals to the cost for moving the container to ODCYs and storing it at ODCYs,

$R(t_0|F_1,S_1) = c_0 + s_0(t_0-F_1)$. Also, from the definition of T_{S2} (= t_0), $R(t_0|0,S_2) = c_0 + s_0t_0$. From these two equalities, it follows $R(t_0|F_1,S_1) < R(t_0|0,S_2)$. And, $R(F_1|F_1,S_1) = 0$ and $R(F_1|0,S_2) = S_2F_1$. Thus, it follows $R(F_1|F_1,S_1) < R(F_1|0,S_2)$. ☐

In the above formulation, it is assumed that the unlimited storage space is available at no additional cost. However, in most of cases, it may be limited and it may be supplied with a cost. Then, the formulation becomes

(P2)

$$\max_{A,F,S,T_s} E(PF(F,S)) = u\int_F^{T_s} S(x-F)f(x)dx - c_sA \tag{4}$$

subject to

$$S(T_s-F) = c_0+s_0(T_s-F) \tag{5}$$

$$uT_s \le A \tag{6}$$

$$S, F, A, T_s \ge 0$$

Property 2: For (P2), F=0 and $A=uT_s$ in an optimal solution.

Proof) The proof will be omitted.

Thus, the problem (P2) becomes

$$\max_{A,S,T_s} E(PF(S)) = u\int_0^{T_s} Sxf(x)dx - c_sA \tag{7}$$

subject to

$$T_s = \frac{c_0}{S-s_0},$$

$$uT_s = A, \tag{8}$$

$$A, S, T_s \ge 0.$$

The problem can be rewritten as follows:

$$\max_S E(PF(S)) = \int_0^{\frac{c_0}{S-s_0}} Sxf(x)dx - c_s\frac{c_0}{S-s_0}. \tag{9}$$

The value of S maximizing $E(PF(S))$ may be found easily by a line search algorithm.

Suppose that the total available space is limited and the whole space is shared by multiple vessels. Let a_{tot} (TEU-days) be the total available during a week and *n* vessels are arriving at the terminal every week. Then, the formulation becomes as follows:

(P3)

$$\max_{F,S,T_{si},A_i} \sum_{i=1}^n E(PF_i(F,S)) = \sum_{i=1}^n \left[u_i\int_F^{T_{si}} S(x-F)f_i(x)dx - c_sA_i\right] \tag{10}$$

subject to

$$S(T_{si}-F) = c_{0i}+s_{0i}(T_{si}-F) \tag{11}$$

$$u_iT_{si} \le A_i, \tag{12}$$

$$\sum_{i=1}^n A_i \le a_{tot}, \tag{13}$$

$$S, F, T_{si}, A_i \ge 0$$

By the same reason in Property 2, F=0 and $u_iT_{si} = A_i$ for all i in an optimal solution. Thus, (P3) can be rewritten as follows:

$$\max_{T_{si},S} \sum_{i=1}^n E(PF_i(S)) = \sum_{i=1}^n \left[u_i\int_0^{T_{si}} Sxf_i(x)dx - c_su_iT_{si}\right] \tag{14}$$

subject to

$$T_{si} = \frac{c_{0i}}{S-s_{0i}}, \tag{15}$$

$$\sum_{i=1}^n u_iT_{si} \le a_{tot}, \tag{16}$$

$$T_{si}, S \ge 0.$$

Relaxing constraint (15), we have the objective function as follows:

$$\sum_{i=1}^{n}\left[u_i \int_0^{T_{si}} Sxf_i(x)dx - c_s u_i T_{si}\right] + \theta\left(a_{tot} - \sum_{i=1}^{n} u_i T_{si}\right), \tag{17}$$

which can be rearranged, after removing the constant term, as follows:

$\sum_{i=1}^{n}\left[u_i \int_0^{T_{si}} Sxf_i(x)dx - (c_s+\theta)u_i T_{si}\right]$ and after replacing T_{is} and removing the constant multiplier, we get

$$\sum_{i=1}^{n}\left[u_i \int_0^{\frac{c_{0i}}{S-s_{0i}}} Sxf_i(x)dx - (c_s+\theta)u_i \frac{c_{0i}}{S-s_{0i}}\right].$$

This problem can be decomposed into n independent sub-problems each of which is to find the optimal solution for the problem as follows:

$$\mathrm{Max}_S E[TPF(S)] = \int_0^{\frac{c_{0i}}{S-s_{0i}}} Sxf_i(x)dx - (c_s+\theta)\frac{c_{0i}}{S-s_{0i}}. \tag{18}$$

E(TPF(S)) can be maximized with respect to S by using a line search algorithm. The optimal value of θ can be obtained by using a line search algorithm.

3 RE-MARSHALING AND ITS IMPACT ON THE SPACE REQUIREMENT

Because outbound containers with the same attributes are loaded onto a vessel consecutively. To speed up the loading operation, outbound containers with the same attributes are stored in the adjacent area in the yard. However, outbound containers arrive at the yard randomly. The space planners in the terminal usually pre-plan the storage layout for outbound containers in advance. When a sufficient space for outbound containers for a vessel becomes available, the planner reserves a specific area in the yard for all the outbound containers. Then, arriving containers are stored in the area according to the layout constructed by the planner, which is called "permanent storage." However, to provide all the required space for future arriving containers, a large amount of space must be pre-allocated, which is not possible unless a large space is available in a terminal. When the space is not enough, one of the most popular strategies is to provide a smaller amount of space in the early stage of container arriving and store arriving containers in a random way, which is called "temporary storage." In this case, a segregated space is not provided to each group of containers and instead containers are mixed in the same area without classification. And then, when the space becomes available later, the layout, which is called a space plan, is constructed by a planner and then the containers arrived so far are moved to the reserved space according the layout, which is called "re-marshaling operation."

When the re-marshaling operation is done in an earlier period, then the number of containers to be re-marshaled is smaller but the full space corresponding to the final layout should be reserved for a longer period of time. If the re-marshaling is done in a later period, then the number of containers to be re-marshaled becomes larger and the full space may be reserved for a shorter period of time. That is, in the former case, a larger amount of space needs to be supplied with a smaller number of containers to be re-marshaled but the latter case represents the opposite case.

This section attempts to analyze the relationship between the re-marshaling time and the size of the reserved area for a vessel and proposes a method to determine them. The problem can be formulated as follows:

(R1)

$$\min_{A,\,T_r} E[TC(T_r, A)] = c_y u \int_{T_r}^{\infty} f(x)dx + c_S A \tag{19}$$

subject to

$$u\int_{T_r}^{\infty}(x-T_r)f(x)dx + uT_r \le A, \tag{20}$$

$$A,\ T_r \ge 0.$$

The first term in the objective function represents the re-marshaling cost which is proportional to the number of containers arriving before the re-marshaling time. The first term in constraint (20) represents the space occupied by containers arriving before the re-marshaling time and the second term is the space occupied by containers after re-marshaling operation.

The second objective term becomes smaller as the value of A becomes lower. Let $G(T_r) = u\int_{T_r}^{\infty}(x - T_r)f(x)dx + uT_r$. Then, $\frac{dG(T_r)}{dT_r}$ = u-u$\int_{T_r}^{\infty} f(x)dx > 0$ for all $T_r > 0$. That is, the left-hand side of the constraint (20) increases as T_r increases. But, the first term of the objective function decreases as T_r increases. Thus, the optimal value of A must satisfy constraint (20) in equality form. That is, $u\int_{T_r}^{\infty}(x - T_r)f(x)dx + uT_r = A$. Thus, this problem becomes

$$\min_{T_r} E[TC(T_r, A)] = c_y u \int_{T_r}^{\infty} f(x)dx + c_s u\left[\int_{T_r}^{\infty}(x - T_r)f(x)dx + T_r\right].$$

Simplifying the objective function,

$E[TC(T_r)/u] = c_y \int_{T_r}^{\infty} f(x)dx + c_s\left[\int_{T_r}^{\infty}(x - T_r)f(x)dx + T_r\right].$

$\frac{dE[TC(T_r)/u]}{dT_r} = -c_y f(T_r) - c_s \int_{T_r}^{\infty} f(x)dx + c_s.$

The optimal value of T_r can be obtained by solving

$$-c_y f(T_r) - c_s \int_{T_r}^{\infty} f(x)dx + c_s = 0. \quad (21)$$

Suppose that the total amount of the space available for outbound containers in a terminal is limited (a_{tot}). If we allocate the space for each vessel proportionally to u_i, which is the amount of containers to be loaded onto vessel i, then, the space to be allocated to each vessel can be determined as follows:

(R2)

$$\min_{A_i,T_{ri}} \sum_{i=1}^{n} E[TC_i(T_{ri}, A_i)] = \sum_{i=1}^{n}\left[c_y u_i \int_{T_{ri}}^{\infty} f_i(x)dx + c_s A_i\right] \quad (22)$$

subject to

$$u_i \int_{T_{ri}}^{\infty}(x - T_{ri})f(x)dx + u_i T_{ri} \le A_i, \quad (23)$$

$$\sum_{i=1}^{n} A_i \le a_{tot} \quad (24)$$

$$A_i, T_{ri} \ge 0.$$

Note that n is the number of vessels per week and a_{tot} is the total available space for outbound containers, A_i is the amount of space allocated to vessel i, u_i is the number of containers for vessel i, and T_{ri} is the re-marshaling time for containers bound for vessel i.

By the same logic for the problem of (R1), $u_i \int_{T_{ri}}^{\infty}(x - T_{ri})f(x)dx + u_i T_{ri} = A_i$. The objective function becomes

$$\sum_{i=1}^{n}\left[c_y u_i \int_{T_{ri}}^{\infty} f_i(x)dx + c_s u_i\left\{\int_{T_{ri}}^{\infty}(x - T_{ri})f(x)dx + T_{ri}\right\}\right]. \quad (25)$$

The constraint (24) of (R2) can be rewritten as follows:

$$\sum_{i=1}^{n} u_i\left\{\int_{T_{ri}}^{\infty}(x - T_{ri})f(x)dx + T_{ri}\right\} \le a_{tot}, \quad (26)$$

$$T_{ri} \ge 0, \text{ for all i.}$$

Relaxing constraint (26), we obtain the following objective function:

$$\sum_{i=1}^{n}\left[c_y u_i \int_{T_{ri}}^{\infty} f_i(x)dx + (c_s+\theta)\mathrm{u}_i\left\{\int_{T_{ri}}^{\infty}(x-T_{ri})f(x)dx + T_{ri}\right\}\right] - \theta a_{tot}.$$

After removing the constant term, the relaxed problem becomes

(RR1)

$$\min_{T_{ri}} \sum_{i=1}^{n}\left[c_y u_i \int_{T_{ri}}^{\infty} f_i(x)dx + (c_s+\theta)\mathrm{u}_i\left\{\int_{T_{ri}}^{\infty}(x-T_{ri})f(x)dx + T_{ri}\right\}\right], \tag{27}$$

$$T_{ri} \geq 0.$$

(RR1) can be decomposed into n independent problems as follows:

(RR2)

$$\min_{T_{ri}} \left[c_y u_i \int_{T_{ri}}^{\infty} f_i(x)dx + (c_s+\theta)\mathrm{u}_i\left\{\int_{T_{ri}}^{\infty}(x-T_{ri})f(x)dx + T_{ri}\right\}\right], \tag{28}$$

$$T_{ri} \geq 0.$$

From (21), we know that the optimal solution of (RR2) is the solution of the equation

$$-c_y f(T_{ri}) - (c_s+\theta)\int_{T_{ri}}^{\infty} f(x)dx + (c_s+\theta) = 0. \tag{29}$$

When the value of θ is given, from (29), the values of T_{ri} s are calculated. Then, the objective value of (28) can be evaluated. Thus, a line search with respect to θ may solve the problem.

4 CONCLUSIONS

This study discussed how to determine the storage charge and the re-marshaling time of outbound containers. When the storage charge is high, the duration-of-stays of containers at the yard of a container terminal become shorter and as a result the space requirement for the storage of outbound containers becomes smaller. It is discussed how to optimize the storage charge for the maximization of the revenue of the terminal operator. It is proved that the optimal free-time-limit for the storage is zero. The re-marshaling is usually done for the saving of the storage space for outbound containers. It is analyzed how the re-marshaling time is related to the space requirement and the optimal re-marshaling time can be obtained for various situations. Considering the trade-off relationship between the total storage space and the re-marshaling time, it is discussed how to determine the space allocation amount for the storage space to each vessel and the re-marshaling time at the same time.

ACKNOWLEDGEMENTS

This work has been partially supported by Korea Maritime Institute through the project entitled "Roadmap for Future Logistics Technologies."

REFERENCES

Castilho, B.D., Daganzo, C.F. (1991), Optimal pricing policies for temporary storage at ports. *Transportation Research Record* 1313, 66-74.

Holguin-Veras, J., Jara-Diaz, S. (1999), Optimal pricing for priority service and space allocation in container ports. *Transportation Research Part B* 33(2), 81-106.

Holguin-Veras, J., Jara-Diaz, S. (2006), Preliminary insights into optimal pricing and space allocation at intermodal terminals with elastic arrivals and capacity constraint. *Networks and Spatial Economics* 6, 25-38.

Kim, K.H. and Kim, K.Y. (2007), Optimal price schedules for storage of inbound containers, *Transportation Research Part B* 41, 892–905

Lee, C.-Y. Lee and Yu, M. (2012), Inbound container storage price competition between the

container terminal and a remote container yard. *Flexible Services and Manufacturing Journal*, on-line available, DOI: 10.1007/s10696-011-9103-6
Taleb-Ibrahimi, M., Castilho, B.D., Daganzo, C.F. (1993), Storage space vs handling work in container terminals. *Transportation Research Part B* 12(1), 13-32.
Watanabe, I. (2001), Container Terminal Planning – A Theoretical Approach. *World Cargo News* (WCN) Publishing, Great Britain, 58-61.
Woo, Y.J. and Kim, K. H. (2011), Estimating the space requirement for outbound container inventories in port container terminals, *Int. J. Production Economics* 133, 293–301.

A NOVEL APPROACH FOR EFFECTIVE SOLUTION OF YARD CRANE SCHEDULING AT SEAPORT CONTAINER TERMINALS

Mehmet Ulaş Koyuncuoğlu[1], Osman Kulak[1], Olcay Polat[1] and Mustafa Egemen Taner[1]

[1]*Department of Industrial Engineering, Pamukkale University, 20070, Denizli, Turkey, ulas@pau.edu.tr, okulak@pau.edu.tr, opolat@pau.edu.tr, metaner@pau.edu.tr*

ABSTRACT

Conditions of competition in container terminals, which are the exit gates of the international networks of the container deployment tools, depend on the efficiency of these terminals. Effective use of the terminal equipments such as cranes in berths and in storage areas and the transporter vehicles between these two areas are the most important factors which affect the efficiency of the terminals. Yard Crane Scheduling (YCS) which are used in the operations generated in the storage areas has an important role in improving the efficiency of the terminals. In this paper, a new approach to free and restrictive deployment’s rules, which is used in yard crane scheduling, is presented and also customer priority and assignment rules are investigated in combination. In this context, for the purpose of minimizing the total completion time of the operations, a mixed-integer linear mathematical model is proposed in order to solve the problems of assigning one or more yard cranes simultaneously and of routing. Results of this model, which is analyzed by using a genetic algorithm based approach, are presented with statistical analysis.

Keywords: Yard Crane Scheduling, Container Terminal, Genetic Algorithm

1 INTRODUCTION

Container transportation has the maximum load-carrying capacity in transnational transportation. Depending on the population growth and global economic policies, container transportation has grown by about 15% each year.

Designed as to carry too much and a wide variety of load at a time, containers, which are reliable, can be piled many times to an area, minimize wastage of goods, are airproof, are lockable, carry/store dangerous goods, are standard transportation/carrying equipments. Container terminals are the junctions of transportation systems in global transportation network. A container terminal, which combines vessels at sea and vehicles on land, is a point where transportation modes meet. A container terminal serves as the temporary storage area between/during the two trips of the containers more than a provider of loading/unloading services for container carriers (Zhang et al., 2003). Services held in a container terminal, transportation/carrying and handling operations together with general carrying equipments used in the terminal are shown in Fig. 1. This study includes the observation of yard cranes which perform stack operations in the storage areas of container terminals.

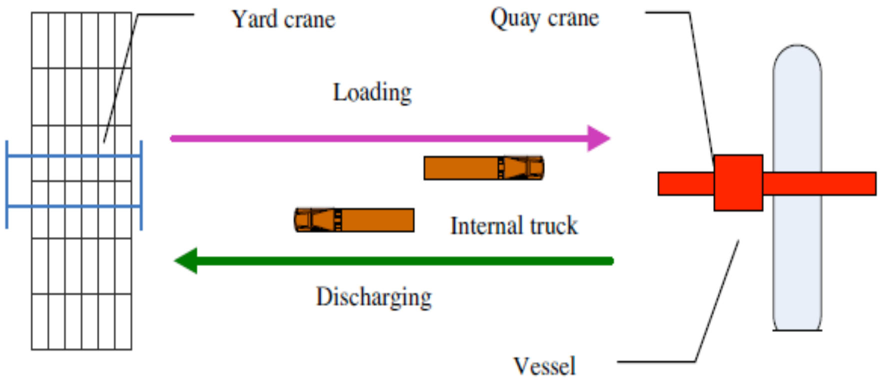


Figure 1. Illustration of Typical Equipments Used in Container Terminals(Yan et al., 2011).

Yard cranes are used to transfer containers in stacking blocks to trucks or container terminals in trucks to stacking blocks (Kim et al. 2003). These equipments move faster than quay cranes due to their hardware futures. Yard cranes especially in automated container terminals move flexibly on a block due to their size. A yard crane which handles the container block works together with the other stacking crane which is on the same block without restricting each other's movements. This flexible structure entails yard cranes and carrier vehicle to be used efficiently. Thus, waiting times are decreased while the amount of containers handled per unit time is increased. In this study, Automatic Stacking Crane (ASC) type yard cranes which exist in automatic container terminals are used.

It is possible to carry out flexible operations owing to the three dimensional motion (X, Y, Z coordinates) of yard cranes, see Fig. 2. Trolleys are the operators of spreader and portal stacking cranes and they move in separate coordinates from each other. In these coordinates, yard crane moves with the portal in Z-coordinate, the spreader in Y-coordinate and the trolley in X-coordinate. Yard cranes move slower during unloading and loading while handling the containers, namely when it is loaded.

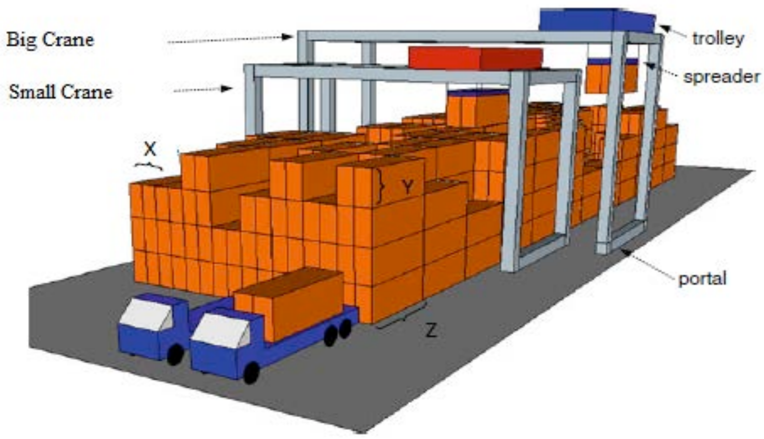


Figure 2. Example of Stacking Block with Two Stacking Crane

Yard cranes are able to carry out operations on the same block according to the “restrictive” and “free” deployment rules. According to the strategies determined by the terminal management, customer priority rule can be applied. The aim of this study is to decide the best strategy looking at the minimum total handling time by scheduling the operations of yard cranes, which works on the container block, with a genetic algorithm software and applying deployment rules and customer priority rules. According to the scheduling methods above, performance of a single or a twin yard crane will be measured. Total handling time of a container block will generate the objective function.

This paper includes the research of different combinations and effective solutions of genetic algorithms and Nearest Neighbor, Lin-Kernighan. Yard crane scheduling problem is dealt and strategies used for scheduling are explained in the second part. After presenting the literature studies about this issue, mixed integer mathematical model is shown with assumptions of the problem. In the third part, heuristic methods used in this study are explained with the reasons of use. Chromosome structure, selection, crossing, mutation methods and termination criteria are explained also. In the fourth part, comparisons of experimental studies and the scenarios, which were designed for YCS problem, of developed models are presented. In the last part, summary of the study is given and conclusion and recommendations are aligned.

2 YARD CRANE SCHEDULING (YCS) PROBLEM

Planning of the process, which starts with the arrival of the containers via carrier vehicles to the storage area by sea or by road and ends with the loading of the containers to the carriers vehicles in the storage area, is known as yard crane scheduling (YCS) in the literature. Although there can be seen some differences according to their types, carrier vehicles generally leave the containers to the points called “transfer points” in the storage area or they carry that containers to a point easy to handle by the yard crane, which is in an isolated in the storage area extending along the block. Usually, loading is done after handling all the containers to be unloaded in the stacking block.

Due to its mechanical properties, a yard crane moves three-dimensionally at a time while unloading the containers, which belong to a customer, over the container block.

Yard cranes are able to maintain their operations on the same block according to the “restrictive” and “free” deployment rules. *Free* deployment is that due to their special size, two or more yard cranes handle the containers by passing through each other while *Restrictive* deployment is that handling the containers in their own area without interfering with each other's areas. It is likely to see free operations of yard cranes in the large automated container terminals in the world more frequently. It is thought that in this way it is possible to carry out more effective operations than the restrictive one. In the free deployment rule, free movements of yard cranes is profitable in terms of time as they can handle the containers which are closer to the yard cranes than their own location. In the restrictive deployment rule, yard cranes can work more effectively as they cannot handle the entire block, namely move a short distance.

YCS problem in the storage area is examined in different ways. Studies generally concentrate on the handling along the blocks by more yard cranes by sharing that blocks. The size of the yard cranes used is usually the same and they restrain operations as they pass through each other. Thus, studies mostly concentrate on the restrictive areas. However, within the frame of these studies, only the linear movement of the yard crane throughout the block is considered. Studies are mostly formulated by mathematic models and solved by any heuristic/meta heuristic algorithm according to the different properties of scheduling problems.

When the studies in the literature are examined, it is obvious that generally in different sized container terminals, focus is on the issues of comparing of restrictive and free yard crane deployment rules and decreasing preparation times of the yard cranes. Conducted studies about yard crane scheduling are mentioned below:

Bish (2003), works on the scheduling of the carrier vehicles in the container terminal and minimization of the total handling time of yard crane in the stacking area where loading/unloading operations are carried out. In order to solve this transfer problem, a heuristic algorithm is developed. Efficiency of this heuristic algorithm is tested by calculating it considering the worst scenarios. Ng and Mak (2005), use branch-bound algorithm to develop yard crane scheduling operations which process slowly and create a bottleneck. Loading/unloading operations are tried to be accelerated by reducing the preparation time. Also, it is proved that branch-bound algorithm is effective and efficient to find lower and upper limits of the operation times. Genetic Algorithm (GA) developed by Matlab 7 and branch-bound algorithm developed by LINGO are compared in terms of yard crane scheduling by Javanshir and Ganji (2008). GA was provided superior result than branch-bound algorithm. Meng et al. (2008), are in the opinion that while yards cranes, which have different length and width, are working together on a block, they work more efficiently as they are passing through each other during the handling of the container blocks. In this study, they try to make loading/unloading operations, which carried out by Double Rail Mounted Gantry Crane, of container blocks more efficient. Solution of the problem is designed by combining Simulated Annealing Algorithm and heuristic. As a result, it is stated that using a combination of these two approaches, rather than in separation, produces more efficient solutions. Mak and Sun (2009), concentrate on reducing the preparation time during loading/unloading operations which are executed by yard cranes. They develop several algorithms which are to minimize the total run time of yard cranes. TS Crossover and TS Mutation algorithms are developed from the combination of GA and Tabu Search (TS). Comparing the proposed algorithms with GA, their cost-effectiveness is calculated. As a result, it is stated that using a combination of TS Crossover and TS Mutation algorithms produces more efficient solutions. Li et al. (2009), carry out the loading/unloading operations in their studies realistically by using a model developed for yard cranes, considering the sharing between the yard cranes. Moreover, it is observed that the suggested model produces effective results. Also, this heuristic algorithm and formulation are flexible enough to be used for some other problems. He et al. (2010), try to develop a novel strategy for yard crane scheduling. With this aim, a dynamic scheduling model is developed for yard cranes by using objective programming. They use heuristic method and Parallel Genetic Algorithm (PGA) in a hybrid way in order to solve the yard crane scheduling problems. Afterwards, a simulation model is developed to evaluate this approach.

When the studies above are examined, it is obvious that calculation is done just by considering the Z-coordinate motion, but not the motion along three coordinates on the container block. In the new method developed in our study, a new approach is represented by integrating GA with the heuristic methods (Nearest Neighbor, Lin-Kernighan). Efficiency of this new approach is proved mathematically as a result of the tests. That the new approach gives efficient results in terms of both solution quality and solution time indicates the originality of the study.

Some assumptions are made in order to find an efficient solution for YCS problem. These assumptions are as follows: (1) A fixed time for preparation, i.e. 5 sec., is defined for each yard crane The order of priority is determined by adding preparation time to operation time before all handling is carried out. (2) All the containers to which handling operations are applied are the size of 20 TEU. (3) Carrying vehicles leave the containers to the points called “transfer point” and again they wait at these points while the yard crane is being unloaded. These points are at the level of Z-coordinate, where the containers stand during unloading, next to the stacking area, under the yard crane. (4) Handling is carried out over just one block. Firstly, containers to be deployed in the block are being handled and then loading is being done.

3 THE PROPOSED HEURISTICS TO SOLVE YCS

Yard crane operations in containers terminals are like machine-product routing/assignment operations. It is a NP-Hard problem to decide in which order and with which yard crane the yard cranes handle the containers. This problem is defined as NP-Hard problem in the literature due to the difficulties encountered in terms of solution (Wenying, 2010).

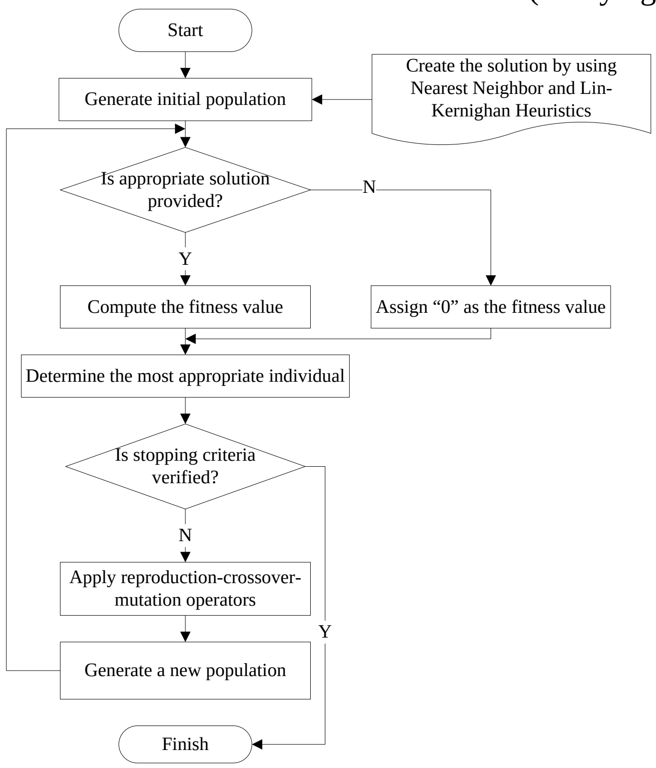


Figure 3. Developed Genetic Algorithm Flowchart

Nearest Neighbor (NN) is chosen to be used in this study considering that yard crane operations are carried out starting from the nearest containers. Lin-Kernighan is integrated in order to develop some methods which decrease the total handling time to optimum and to provide randomness while GA operators are being applied. Each solution obtained from NN is applied to L-K and GA operators are applied in case of improving. 25% of the initial population is produced by using NN and L-K. On the other hand, the best solution value is tried to be found by being produced by Genetic Algorithm (GA) and applying GA operators for the rest 75%. The flow chart of GA is shown in Fig. 3.

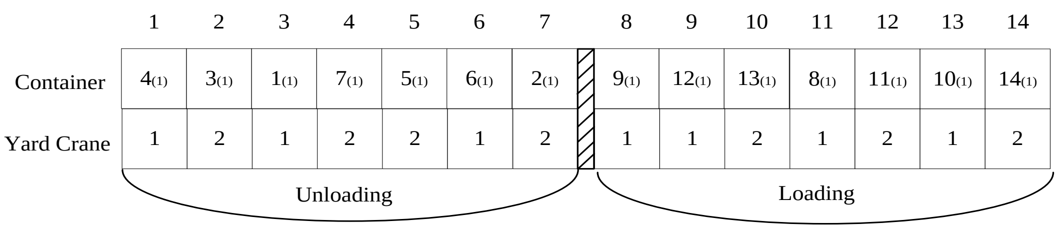


Figure 4. GA Chromosome Structure Used In the Study

As it is seen in Figure-4, chromosome is double-layer. While genetic algorithm operators are being applied, new individuals are being produced by applying those operators to the genes, which represent unloading and loading operations, separately.

In the sample chromosome demonstration above, the first 7 genes represent the 7 containers to be unloaded while the next 7 genes represent the containers to be loaded. In the storage area in the container terminal, firstly loading and then unloading operations should be carried out because of the size of the container block. Alignment in gene demonstration indicates in which order the containers are handled with respect to time and yard crane indicates by which yard crane the container is handled. For example, $4_{(1)}$ gene part shows that 4^{th} container belongs to the 1^{st} customer. In the double-decker structure of the gene, the gene in the bottom represents yard crane. Before each container is handled, 5 seconds of preparation and delay time is defined in the system.

4 EXPERIMENTAL STUDIES

The heuristic model developed in this study is tested for different strategies, which are expressed below, of container terminals. Obtained results are examined statistically.

Single Yard Crane-Priority Customer Group (SYC-PCG): It indicates that only one yard crane works on a container block. It is the strategy of handling the customer group, which the containers belong to, according to the determined handling order.

Single Yard Crane-Single Customer (SYC-SC): It indicates that only one yard crane works on a container clock. It is the strategy of handling of the customer group, which the containers belong to, without considering the order. Because there is no order, it is considered as if there was just one customer.

Restrictive Yard Crane-Priority Customer Group (RYC-PCG): It indicates that two yard cranes work together on a container block. Yard cranes share the area on the basis of Z-coordinate of the block. Sizes of the yard cranes might be different and also they cannot carry out handling operations out of the block area reserved for them. It is the strategy of handling the customer group, which the containers belong to, according to the determined handling order.

Restrictive Yard Crane- Single Customer (RYC-SC): It indicates that two yard cranes work together on a container block. Yard cranes share the area on the basis of Z-coordinate of the block. Sizes of the yard cranes might be different and also they cannot carry out handling operations out of the block area reserved for them. It is the strategy of handling of the customer group, which the containers belong to, without considering the order. Because there is no order, it is considered as if there was just one customer.

Free Yard Crane-Priority Customer Group (FYC-PCG): It indicates that two yard cranes work together on a container block. Yard cranes can carry out the handling operations over the whole block by passing through each others due to the advantage of their sizes. It is the strategy of handling the customer group, which the containers belong to, according to the determined handling order.

Free Yard Crane- Single Customer (FYC-SC): It indicates that two yard cranes work together on a container block. Yard cranes can carry out the handling operations over the whole block by passing through each others due to the advantage of their sizes. It is the strategy of handling of the customer group, which the containers belong to, without considering the order. Because there is no order, it is considered as if there was just one customer.

According to the strategies of the terminal, customer priority rule can be applied. This prioritization is used for both loading and unloading operations.

Problems are solved in order to determine the strategy which gives the shortest time to the fleet sizes of 30 and 60 containers for both single and twin yard cranes on a stacking block (with restrictive-free deployment rules). Two different solution methods were applied. These are simple GA and GA integrated with constructive (NN)-improving (L-K) heuristics. Size of the container block is 5x6x20 (600 containers) and the containers to be loaded or unloaded do not overflow these sizes. There are 300 containers in the existing stacking block. Among 300 containers, 15 unloadings and 15 loadings for the fleet size of 30 containers; 30 unloadings and 30 loadings for the fleet size of 60 containers are done. These containers may belong to different customers due to the differences in the strategies developed. Firstly, containers to be unloaded are determined and prepared for the test data. In order to restrictive yard crane deployment strategy to work efficiently, containers are located on the block as much randomly as possible. After the unloading operation is completed, yard crane tries to leave the container from the location where it is taken from, to the free coordinate which is the closest/the most appropriate.

The developed heuristics are coded using Java programming language and executed on an Intel Core 2 Duo 1.6 GHz processor with 8GB RAM. Experiments are done with the help of the interface by loading the properties of the container block, range of motion and speed of the yard crane, data of the containers (its location on the stacking block, which customer it belongs to and so on.) and parameters of genetic algorithm to the program. Moreover, it is possible to determine a new approach strategy from the interface of the program (Nearest Neighbor + Lin-Kernighan). Termination criterion in GA is determined as the average of fitness value (F_{avg}) to be close/equal (≈% 0) to the best solution (F_{best}). GA parameters, which are used generally and proved to produce good solutions mathematically, are used in our study as a result of literature reviews. Randomness is provided by doing 5 experiments for the combination of each kind of the parameter.

In the container list prepared for the experimental study (test data), locations of the containers to be loaded and unloaded and which customer they belong to are determined randomly. 3840 experiments are carried out by using the GA parameters separately for each strategy. 3840X2=7680 experiments are done for the size of 30 and 60 containers by using all the strategies. On the other hand, 7680x6=46080 experiments are done on the yard crane by applying the deployment rules and priority rules.

In the new approach developed for the SCS problem, scenarios are tried to be optimized also in terms of methods used. Problem is tried to be solved by two different approaches. In the first approach, problem is solved by the developed GA (to produce 100% of the initial population randomly by genetic operators) and in the second approach, solution is produced with a certain extent of NN structural heuristic (25% of NN, 75% of GA) for the initial population. The total number of experiments, with the approaches explained above, is 46080x2=92160. The best fitness value belongs to the experiments are shown in Table 1. Experiments at issue are carried out as the initials of the software developed. It is predicted that the experiment results can be improved.

Table 1. The Best Fitness Values of Solution (Second)

Methods		GA		GANNLK	
Number of Container		30	60	30	60
Deployment Strategies	RYC-PCG	2657,57	5171,74	2574,62	5068,2
	RYC-SC	2646,4	5142,43	2542,23	5028,65
	FYC-PCG	3630,27	6448,16	3347,54	6186,08
	FYC-SC	3614,8	6364,09	3307,88	6105,47
	SYC-PCG	3669,5	6655,98	3573,1	6287,91
	SYC-SC	3614,28	6650,9	3514,66	6233,37

The best solution values in terms of the yard crane strategies in the table above are obtained by aggregating the total handling time of two cranes.

Parameters which provide the best solutions for the new approach and the strategies is defined as population size to be 10 times of the number of the containers, ratio of the elite to be 10%, Rolette Wheel technique to be the chromosome selection technique, Position-Based crossing to be the crossing method, ratio of crossing to be 90%, Right Rotation to be the mutation method, ratio of decreasing the mutation is to be 25%, standard deviation coefficient (c) to be 3, and termination ratio to be 0. These values are determined with the joint statistical significance for the sizes of 30 and 60 containers and for the methods.

When the experiment results are analyzed statistically, it is obvious that there is a significant difference between GA and GANNLK in terms of the quality of the solution. GANNLK approach provides a better fitness value statistically compared to GA.

When the strategies are analyzed statistically, it can be seen that RYC-SC strategy is better than the others. In other words, when two yard cranes are working according to the restrictive area plan on a stacking block in the container terminal, it is possible to handle the containers to be handled. On the other hand, it is thought that although there is not a significant difference between RYC-SC (5028,65) strategy and RYC-PCG (5068,2) strategy, there will be a significant difference for bigger fleet sizes. That the restrictive yard crane deployment strategy produces significantly better results than the others can be explained by the more restrictive motion of the yard crane which moves to handle the containers in the stacking crane. The reason is that a yard crane moves more on the same sized stacking block.

When the strategies developed are examined (for the fleet size of 60 containers), with the best solution value (the objective function) 5028,65 second, RYC-SC is determined as the best strategy, see Fig. 5.

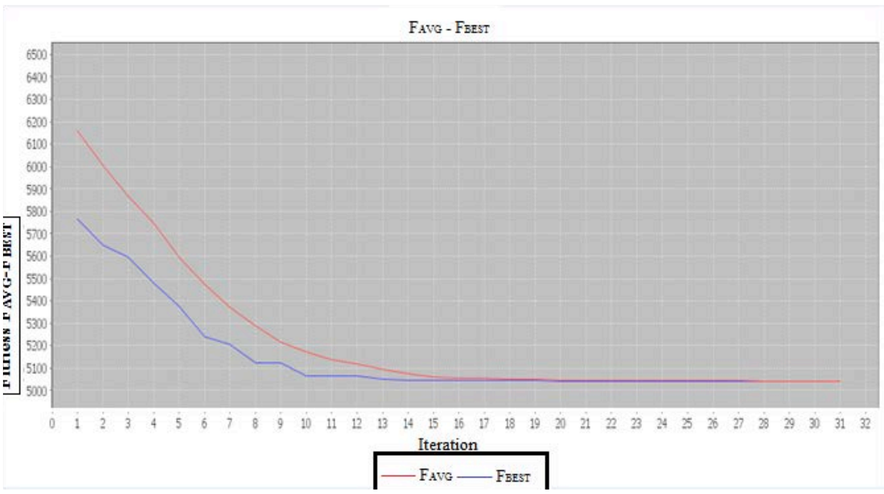


Figure 5. According to RYC-SC Strategy F_{avg}-F_{best} Fitness Values

5 CONCLUSION

In this study, the handling strategies of the yard cranes which handle the containers are emphasized. Yard crane deployment rules and customer priority rules, according to their time of handling and applying integrated, are analyzed by software which is developed by using Genetic Algorithm, Nearest Neighbor and Lin-Kernighan heuristic methods. In order to decide the strategy which provides the least total handling time, fitness value which is the objective function, is considered as the basis.

According to the results, restrictive area dispatching rule with twin cranes and not using the customer priority are the identified as the strategy which provides the best solution for a randomly determined fleet size of containers. When the methods used while comparing the strategies are examined, again the objective function is considered as the basis for the evaluations. It is statistically proved that better results are provided with GANNLK when GA operators are applied to the chromosomes some of the initial population of which is created by NN and improved by L-K improvement heuristic compared to the method which is started with only the random solutions of GA.

It is proved that dispatching rules of the yard cranes and customer priority rule are effective on the container handling time. This study will go on with the improvement of the GA solution times and research on the solution by using different methods (e.g. Tabu Search) of the YCS problem strategies.

REFERENCES

Kim K. H., Lee K. M., Hwang H. (2003), “Sequencing Delivery and Receiving Operations For Yard Cranes In Port Container Terminals”, Int. J. Production Economics 84: 283–292.

Bish E. K. (2003), “A Multiple-Marine-Constrained Scheduling Problem In a Container Terminal”, European Journal of Operational Research 144, 83–107.

Zhang C., Liu J., Wan Y., Murty K. G., Linn R. (2003), “Storage space allocation in container terminals”, Transportation Research Part B, No: 37, 2003, 883-903.

Ng W. C. and Mak K. L. (2005), “Yard Crane Scheduling in Port Container Terminals”, Applied Mathematical Modeling 29, 263–276.

Javanshir H., Ganji S.R.S. (2008), “Yard Crane Scheduling In Port Container Terminals Using Genetic Algorithm”, J. Ind. Eng. Int., 6, 39-50.

Meng, Q., Cao Z., Lee D. H. (2008), Deployment Strategies of Double-Rail-Mounted Gantry Crane Systems For Loading Outbound Containers in Container Terminals, Int. J. Production Economics, 115, 221– 228.

Mak K.L. and Sun D., (2009), “Scheduling Yard Cranes in a Container Terminal Using a New Genetic Approach”, Engineering Letters, 17:4, 1-7.

Li W., Wu Y., Petering M. E. H., Goh M., Souza R., (2009), “Discrete Time Model and Algorithms For Container Yard Crane Scheduling”, European Journal of Operational Research 198, 165–172.

Wenying Y., Junqing S., Fenglian L., Peng Y., Mei H., Meiling F., (2010), “A Novel Multi-RTGC Scheduling Problem Based on Genetic Algorithm”, 978, IEEE, 1-6.

He J., Chang D., Mi W., Yan W., (2010), “A Hybrid Parallel Genetic Algorithm for Yard Crane Scheduling”, Transportation Research Part E 46, 136–155.

Yan W., Huang Y., Chang D., He J., (2011), “An investigation into knowledge-based yard crane scheduling for container terminals”, Advanced Engineering Informatics, 25 462-471.

THE DIRECT SHIP-TO-SHIP CONTAINER TRANSSHIPMENT PROBLEM AT A MARITIME TERMINAL

M. Flavia Monaco[1] and Marcello Sammarra[2]

[1]*Dipartimento di Elettronica, Informatica e Sistemistica – Università della Calabria, Via P. Bucci 41-C, 87036 Rende – Italy, monaco@deis.unical.it*

[2]*Istituto di Calcolo e Reti ad Alte Prestazioni – Consiglio Nazionale delle Ricerche, Via P. Bucci 41-C, 87036 Rende – Italy, sammarra@icar.cnr.it*

ABSTRACT

Transshipment operations at a maritime container terminal are usually implemented according to the ship-yard-ship cycle. The sojourn time of the containers in the yard is in general sufficiently long to guarantee that the loading and discharging operations can be scheduled independently. Terminal planners are recently considering the feasibility of a new operational modality, called *live connection*. In this modality a discharged container is immediately transshipped to the outgoing vessel, completely skipping the yard storage phase. Live connection assumes a partial overlapping of the berthing time windows of the involved ships, whose discharging/loading phases are no longer independent. In this paper we introduce the Direct ship-to-ship Container Transshipment Problem (DCTP) and derive a Mixed Linear Integer formulation.

Keywords: quay crane scheduling, stowage plan, transhipment

1 INTRODUCTION

The conventional transhipment flow of containers at a maritime terminal follows the quay-yard-(yard)-quay cycle, where the yard-to-yard movements concern possible housekeeping operations. The yard is then the core of the whole terminal system, because all the containers flowing through the terminal have to be temporarily stored there. The storage allows to decouple in time the import flow from the output flow. Therefore the discharging and loading operations are independent and, for this reason, they can be planned and scheduled separately. However, due to its limited capacity, the yard is also a critical resource for the terminal management, which aims at maximizing the throughput of the terminal. This can be achieved by reducing the storage time of the containers (container dwell-time), so as to minimize possible yard congestions. On the other hand, the storage of containers represents a relevant cost for the shipping line companies. This amounts to say that reducing the container dwell-times is a common objective for the two main operators of the transhipment market.

In this work we consider the Direct Container Transhipment Problem (DCTP) between vessels. The study has been stimulated by the management of the Gioia Tauro Container Terminal, in the South of Italy, which recently experienced this new operational modality. We consider the case of two vessels, simultaneously moored at not necessarily adjacent berths, given that sufficient terminal equipments (machines and operators) have been previously allocated. We assume that some of the containers carried by the vessels have to be directly transhipped from the one to the other, while the remaining containers, to be loaded and discharged, follow the conventional transhipment modality. Our aim is to schedule all the unloading and loading operations, minimizing the service time of both vessels, while reducing, as far as possible, the temporary storage of the direct transhipped containers.

It should be clear that in the direct transhipment modality the unloading and loading operations are no longer independent and the related scheduling processes are concurrent: the same container represents two dependent tasks, to be executed by different machines (quay cranes) operating on different vessels, linked by a strict precedence relationship. The strong interaction between the unloading and loading operations of containers involved in the direct transhipment, can cause a sharp deterioration of the Gross Crane Productivity. Therefore suitable planning models must be adopted.

We propose a Mixed Integer Linear Model for the DCTP. It integrates two main decisions:

1. determining the working sequences for all the quay cranes operating on the two vessels;
2. assigning a stowage position to all the containers directly transhipped.

As mentioned above, the objective function to minimize is a linear combination of the makespan of the two vessels and the total time the direct transhipped containers spend in the yard buffers.

The paper is organized as follows. In Section 2 we present a literature review. The mathematical model for the DCTP is presented and analyzed in Section 3. Section 4 is devoted to the conclusions.

2 LITERATURE REVIEW

The direct transhipment modality is, in some way, similar to the Cross Docking policy at a distribution terminal in a logistic network (Boysen et al, 2010). The main analogy between them is the common need of synchronizing arrival and departure sequences of the carriers. On the other hand, they significantly differ because in the distribution centres the unit load is first unpacked and then delivered; moreover in the direct container transhipment the stowage positions of the containers induce strong precedence constraints that cannot be ignored.

The effectiveness of the direct transhipment of containers between different means of transport has been investigated in (Alicke, 2002) for the case of rail hubs, in (Lee *et al,* 2006) for the case of trains and trucks, and in (Blumenhagen, 1981) for the case of mother ships and barges.

To the best of our knowledge, the DCTP is new in the literature concerning the maritime container terminal logistics. We found only a paper that refers to the direct ship-to-ship transhipment, but in a very restricted and unusual context. The authors consider a one-way direct transhipment between a mother vessel and a feeder, which “moor at the same berth and utilize the same handling equipment” (Lian *et al*, 2011). Some researchers have recently studied an analogous problem in the case of mobile (or offshore) harbours, which are floating platforms equipped with portal cranes and are used to perform unloading and loading operations in the open sea (Nam and Lee, 2012).

3 MATHEMATICAL MODEL

3.1 Terminology, useful concepts and notation

In this subsection we focus on some basic concepts that will be used throughout the paper. Containers are stowed in a containership into single cells called slots. Each slot can be used to stow a one TEU container. A slot, or equivalently the position of the container inside the containership, is identified by three coordinates: *bay, row, tier*. The bay is the longitudinal coordinate of a slot, while the row and the tier represent, respectively, the transversal and vertical ones. Two adjacent slots, sharing the same row and tier coordinates, are used for storing a two TEUs container. In this case the bay coordinate is also referred as *main bay*. The bay coordinates take positive increasing numbers from the bow to the stern. In order to differentiate main bays from bays the following coding is adopted: the coordinate of the main bay is the even number *b*, while the corresponding two bays are indexed by the odd numbers $b-1$ (*F-bay*) and $b+1$ (*A-bay*), see Figure 1.

The arrangement of the containers within the slots (that is the stowage plan) is done on the basis of the container class. A class is a set of container attributes like: the size, the weight category (light, medium, heavy), the type (reefer, open top), the load (dangerous, perishable) and the port of destination (POD). Containers belonging to the same class (homogeneous containers) are stored into contiguous slots in the same bay. We define task a set of homogeneous containers. For example in Figure 2, two container classes, namely CL_1 and CL_2, and as many tasks are shown. Tasks can also be classified by the operation type they require. Therefore there are loading and discharging tasks.

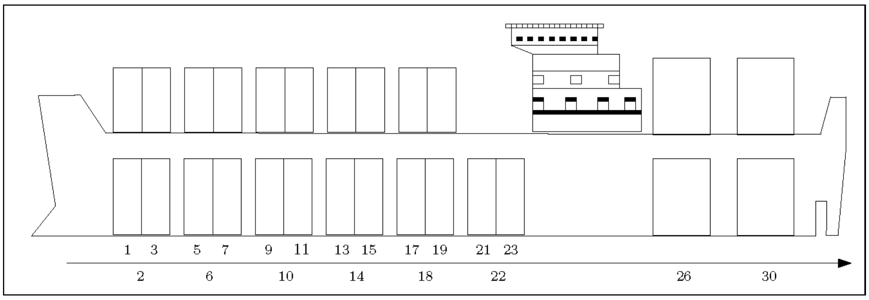


Figure 1. The bay structure of a containership

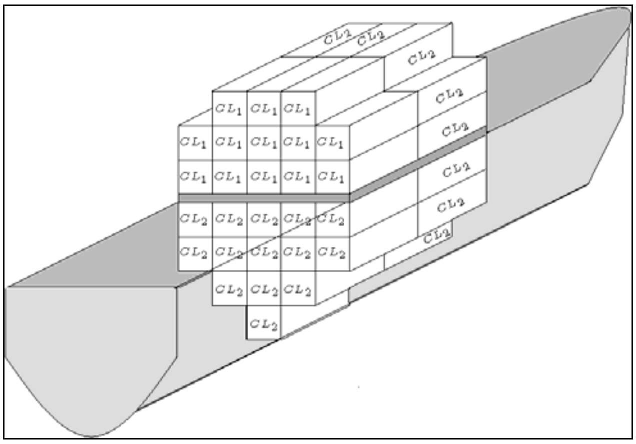


Figure 2. The pre-stow plan of a bay

We now introduce our main notation. Let $V=\{A,B\}$ be the two vessels involved in the live connection. For each $v \in V$ we define:

- a^v the arrival time of v;
- $\pi^v \in [0,1]$ the priority coefficient of v;
- Ω^v the set of loading/unloading operations to be performed on v, also called the task set of the containership v;
- Q^v the set of quay cranes operating on v;
- H^v the set of the bays of the containership *v;*
- Θ^v the set of slots available for stowing containers directly transhipped to the containership v.

Each task set Ω^v consists of three disjoint subsets $\Omega^v = G^v \cup D^v \cup L^v$. G^v is the set of groups of containers, to be loaded and/or discharged, following the conventional transhipment flow; D^v is the set of single containers discharged from v and directly transhipped to the other ship; analogously, L^v is the set of containers directly transhipped from the other ship and to be loaded into the ship v. Each task $i \in \Omega^v, \forall v \in V$ is characterized by a processing time p_i and a bay coordinate $b(i)$, which is an input datum for containers in $G^v \cup D^v$, while it has to be decided for the containers in L^v. For each container $i \in L^v$, γ_i denotes its class.

For each ship, some temporal restrictions on task pairs are defined. They impose either precedence or non-simultaneity constraints on the processing of the tasks. The former results from the operation the tasks require, as well as from their relative position within the same bay, as it can be retrieved from the stowage plan of the vessels. Generally speaking, at a given bay, discharging always precedes the loading operation and discharging the deck precedes the discharge of containers from the hold. The loading operations follow the opposite deck/hold precedence. The non-simultaneity relationships are useful to avoid that tasks too close in space, for example tasks located in adjacent bays, are processed simultaneously by different cranes. This is because the cranes must keep a safety distance all the time they work on a containership. This set of constraints is also known as non-interference constraints between cranes The sets of precedence and non-simultaneity relationships will be deeply investigated in the next subsection.

As regards the cranes, we assume that they are identical machines and are aligned along the quay where the ships are berthed (that is no setup movements are needed). The cranes $k \in Q^v$ are not allowed to span all over the ship v, but they operate only within a predefined range of bays $[s_k, f_k]$, and are released at the time r_k. In particular, their starting position are denoted as l_k^0, $s_k \le l_k^0 \le f_k, k \in Q^v, v \in V$ and such starting positions induce the crane ordering: if $l_k^0 < l_h^0$, then $k < h$. They move along the quay with a speed of s meter per second. Therefore the time, in seconds, a generic crane takes to reach the bay b_2 starting from the bay b_1, is $t_{b_1 b_2} = 6 s^{-1} |b_1 - b_2|$, where 6 is approximately the length in meters of a bay; similarly, the time the crane k needs for reaching the bay b of the first task it has to handle is given by $t_{kb}^0 = 6 s^{-1} |l_k^0 - b|$. For a given bay $b \in H^v$, we indicate by $Q(b) = \{k \in Q^v | b \in [s_k, f_k]\}$ the set of cranes that can operate the bay b.

As regards the slot positions, the elements of each set Θ^v are actually obtained by mapping the physical slot coordinates (*bay, row, tier*) in to a single index ϑ. We assume that $|\Theta^v| = |L^v|$; we define γ_ϑ to be the class of the slot ϑ, as indicated by the stowage plan, and b_ϑ to be the bay coordinate of the slot ϑ. Therefore, if $i \in L^v, \Theta_i^v = \{\vartheta \in \Theta^v | \gamma_i = \gamma_\vartheta\}$ and $H_i^v = \{b \in H^v | b = b_\vartheta, \vartheta \in \Theta_i^v\}$ are, respectively, the sets of slots and bays compatible with the container i, that is where the container can be stowed. We note that H_i^v can be defined also for the tasks $i \in G^v \cup D^v$. In this case we have simply $H_i^v = \{b(i)\}$.

Finally, $\Theta^v(b,i) = \{\vartheta \in \Theta_i^v | b_\vartheta = b\}, i \in L^v, b \in H_i^v, v = A, B$ represents the set of slots compatible with the container i in the bay b.

3.2 Pre-processing

In order to formulate the problem, we consider two containerships *A* and *B*, whose structure, in terms of number of bays, and cranes assigned to each of them, are given. Without loss of generality we assume that

1. both vessels *A* and *B* are berthed port-side;
2. the berthing time windows of the two vessels overlap, at least partially;
3. all the containers are of the same type (one TEUS);
4. the bay coordinates take consecutive integer values from the bow to the stern. The lowest bay coordinate is 1 for both vessels, while the highest ones are l_A^F and l_B^F.

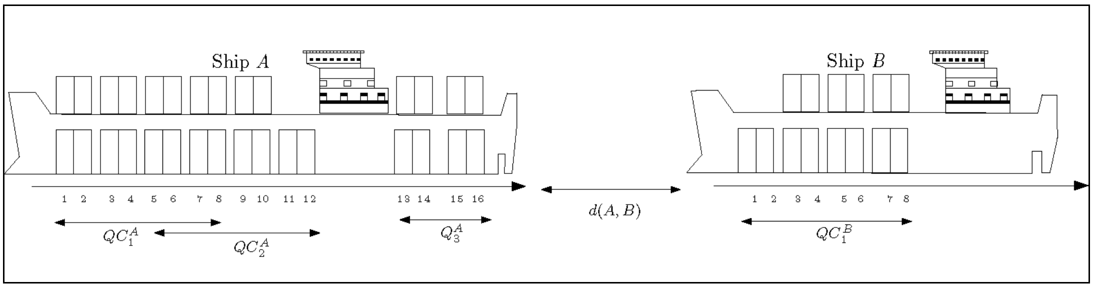


Figure 3. The virtual vessel

As in Figure 3, we assume that the distance between the stern of *A* and the bow of *B*, that is the inter-ship distance $d(A,B)$, is not less than the clearance distance that two adjacent berthed vessels must keep. The basic idea is to merge the two containerships into a virtual vessel $\tilde{V}$ with task set $\Omega = \Omega^A \cup \Omega^B$ and crane set $Q = Q^A \cup Q^B$. By this way the DTCP can be formulated as a Quay Crane Scheduling Problem on the virtual vessel, with side constraints, related to the additional decisions on the stowage of the containers directly transhipped between the two vessels. Hence we will make use of well-known mathematical models for the QCSP, enriching these basic models so as to represent the peculiarities of the DCTP. The reader interested to the QCSP can refer, for example, to (Bierwirth and Meisel), 2009), (Kim and Park, 2004), (Moccia *et al*, 2006) (Monaco and Sammarra, 2011). Here we want just to underline that all the previously cited papers consider the precedence constraints between tasks in a similar way, while the modelling of the interference constraints between cranes differ from each other.

We now describe the precedence and then non-simultaneity relationships between pairs of tasks of $\tilde{V}$. A first set precedence relations is denoted by $\Phi = \Phi^A \cup \Phi^B \cup \bar{\Phi}$, where

$$\Phi^v = \left\{ (i,j) | i \to j,\ i,j \in G^v \cup D^v \right\} \quad v = A,B \tag{1}$$

$$\bar{\Phi} = \left\{ (i,j) | i \in D^A, j \in L^B \vee i \in D^B, j \in L^A \right\} \tag{2}$$

In particular each Φ^v consists of precedence relationships between containers or groups of containers belonging to the same vessel. For such tasks the stowing position and the type of operation they require is known, as for the classic QCSP. Therefore Φ^v can be computed by the stowage plans of the involved vessels. The set $\bar{\Phi}$ results, instead, by considering each container to be directly transhipped as two different tasks, that is a discharging and a loading task, belonging to different vessels. Further precedence relationships for a given vessel v are needed to guarantee that the directly transhipped containers will be loaded according to the decided stowage plan. To this aim we define

$$\Phi_1^v = \left\{ (i,\vartheta) | i \to \vartheta, i \in G^v \cup D^v, \vartheta \in \Theta^v \right\} \quad v = A,B \tag{3}$$

$$\Phi_2^v = \left\{ \left(\vartheta, i \right) | \, \vartheta \to i, \, \vartheta \in \Theta^v, \, i \in G^v \right\} \quad v = A, B \tag{4}$$

$$\Phi_3^v = \left\{ \left(\vartheta_1, \vartheta_2 \right) | \, \vartheta_1 \to \vartheta_2, \, \vartheta_1, \vartheta_2 \in \Theta^v \right\} \quad v = A, B \tag{5}$$

The sets in (3) and (4) define the precedence relationships between a task whose stowage position is known and the container that will be stowed into the ship-slot ϑ. For example, in (3), i could be a task to be discharged from the deck of the bay $b(i)$, while ϑ a ship-slot belonging to the same bay but located into the hold, where a generic compatible container $j \in L^v$ can be stowed. The sets in (5) define precedence relationships between pairs of ship-slots due to their relative positions within the same bay. As a consequence they will induce, at runtime, also a set of precedence relations on the containers that will be stowed there.

The non-simultaneity relations are needed to avoid interferences between cranes (see also Section 3.1). In the classical models for the QCSP, they are defined as pairs of tasks that cannot be processed at the same time by adjacent cranes, since they are located in too close bays:

$$\Psi = \left\{ \left(i, j \right) | \, | \, l_i - l_j \, | \le \delta \right\} \tag{6}$$

where l_i, l_j are the bay coordinates of tasks i and j, and δ is the safety distance between two adjacent cranes, expressed in number of bays.

Here, instead, we directly consider the couples of bays which cannot be operated simultaneously by different cranes. We define:

$$\Psi^v = \left\{ \left(b_1, b_2 \right) | \, b_1, b_2 \in H^v, \, b_1 < b_2, \, b_2 - b_1 \le \delta \right\} \quad v = A, B \tag{7}$$

The sets (7) will allow to properly impose the non-simultaneity constraints between each pair of tasks $i, j \in \Omega^v$ located, respectively, in the bays b_1, b_2, even for the loading tasks whose stowage bay must be decided by the model (see Section 3.3).

As disclosed in (Bierwirth and Meisel, 2009), the set Ψ is not sufficient to model the interferences between cranes in the classic QCSP. To overcome this issue, the authors have introduced a minimum time span that has to be kept between the processing of any two tasks, say i and j, by two cranes h and k, handling, respectively, i and j:

$$\Delta_{ij}^{hk} = \begin{cases} 6 \, s^{-1} \left(l_i - l_j + \delta_{hk} \right) & h < k, i \ne j, l_i > l_j - \delta_{hk} \\ 6 \, s^{-1} \left(l_j - l_i + \delta_{hk} \right) & h > k, i \ne j, l_i < l_j + \delta_{hk} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\delta_{hk} = \left(\delta + 1 \right) \left| h - k \right|$ represents the smallest allowed distance between the cranes h and k. Then in (Bierwirth and Meisel, 2009) all the possible combinations of tasks and cranes causing interferences are represented by the set $\hat{\Delta} = \left\{ \left(i, j, h, k \right) | \, \Delta_{ij}^{hk} > 0 \right\}$. It is easy to see that if $\left(i, j \right) \in \Psi$, then $\left(i, j, h, k \right) \in \hat{\Delta}$ for each possible pair of cranes. In this sense $\hat{\Delta}$ is a generalization of Ψ.

Furthermore we note that if $\Delta_{ij}^{hk} > 0$, that is if the tasks i and j, lying in the bays l_i and l_j, cannot be processed at the same time by the cranes h and k, then no other pair of tasks lying in the same bays do. In other words, the non-simultaneity is a feature of the bays, rather than of the tasks. Therefore, in our notation, the above definition (8) can be reformulated as:

$$\Delta_{b_1b_2}^{hk}(v)=\begin{cases}6\,s^{-1}\left(b_1-b_2+\delta_{hk}\right) & h<k, b_1>b_2-\delta_{hk}\\ 6\,s^{-1}\left(b_2-b_1+\delta_{hk}\right) & h>k, b_1<b_2+\delta_{hk}\\ 0 & \text{otherwise}\end{cases} \qquad h,k\in Q^v, b_1,b_2\in H^v, v=A,B \tag{9}$$

or, in a more compact form:

$$\Delta_{b_1b_2}^{hk}(v)=6s^{-1}\max\left\{(\delta+1)(k-h)-(b_2-b_1),0\right\} \qquad h,k\in Q^v, b_1,b_2\in H^v, v=A,B \tag{10}$$

As a consequence the set

$$\hat{\Delta}(v)=\left\{\left(b_1,b_2,h,k\right)|\,\Delta_{b_1b_2}^{hk}(v)>0\right\} \quad v=A,B \tag{11}$$

represents all the combinations of bays and cranes causing interferences, and, as in the QCSP, $\left(b_1,b_2\right)\in\Psi^v\Rightarrow\left(b_1,b_2,h,k\right)\in\hat{\Delta}(v)\ \forall h,k\in Q^v$.

3.3 The Model

We define the following decision variables:

- $x_{ijk}\in\{0,1\}\ \forall i,j\in\Omega^v, k\in Q^v, v=A,B$, where $x_{ijk}=1$ if the tasks i and j are processed consecutively by the crane k; 0 otherwise;
- $z_{ij}\in\{0,1\}\ \forall i,j\in\Omega^v, v=A,B$, where $z_{ij}=1$ if the task i is completed before the processing of j starts; 0 otherwise;
- $y_{i\vartheta}\in\{0,1\}\ \forall i\in L^v, \vartheta\in\Theta_i^v, v=A,B$, where $y_{i\vartheta}=1$ if the container i is stowed in the slot ϑ; 0 otherwise;
- $\alpha_{ib}\in\{0,1\}\ \forall i\in\Omega^v, b\in H_i^v, v=A,B$, where $\alpha_{ib}=1$ if the task i is stowed in a compatible slot of the bay *b*; 0 otherwise. As $H_i^v=\{b(i)\}, \forall i\in G^v\cup D^v$, $\alpha_{ib(i)}=1$, $\alpha_{ib}=0\ \ \forall b\neq b(i)$ are input data;
- $c_i\geq 0\ \forall i\in\Omega^v, v=A,B$, the completion time of the task i;
- $\sigma_{ij}\geq 0\ \forall(i,j)\in\overline{\Phi}$, the time needed to transfer a directly transhipped container from the discharging to the loading bay;
- $w^v\geq 0$, $v=A,B$, the makespan of the vessel v.

$$\min \quad \pi^A w^A+\pi^B w^B+\beta\sum_{(i,j)\in\overline{\Phi}}\left(c_j-p_j-\sigma_{ij}-c_i\right) \tag{12}$$

$$\sum_{j\in\Omega_T^v} x_{0jk}=1 \qquad k\in Q^v, v=A,B \tag{13}$$

$$\sum_{i\in\Omega_0^v} x_{iTk}=1 \qquad k\in Q^v, v=A,B \tag{14}$$

$$\sum_{k\in Q^v}\sum_{j\in\Omega_T^v} x_{ijk}=1 \qquad i\in\Omega^v, v=A,B \tag{15}$$

$$\sum_{j\in\Omega_T^v} x_{ijk}-\sum_{j\in\Omega_0^v} x_{jik}=0 \qquad i\in\Omega^v, k\in Q^v, v=A,B \tag{16}$$

$$\sum_{i\in L^v} y_{i\vartheta}=1 \qquad \vartheta\in\Theta_i^v, v=A,B \tag{17}$$

$$\sum_{b\in H_i^v}\alpha_{ib}=1 \qquad i\in L^v, v=A,B \tag{18}$$

$$\sum_{\vartheta\in\Theta^v(i,b)} y_{i\vartheta} = \alpha_{ib} \qquad i \in L^v, b \in H_i^v, v = A,B \quad (19)$$

$$\sum_{k\in Q(b)} \sum_{j\in\Omega^v} x_{ijk} \geq \alpha_{ib} \qquad i \in \Omega^v, b \in H_i^v, v = A,B \quad (20)$$

$$c_i - p_i \geq a^v \qquad i \in \Omega^v, v = A,B \quad (21)$$

$$r_k - c_j + 6s^{-1} \sum_{b\in H_j^v} \alpha_{jb} t_{l_k^0 b} + p_j \leq M\left(1 - x_{0jk}\right) \qquad j \in \Omega^v, k \in Q^v, v = A,B \quad (22)$$

$$c_i + \left(\alpha_{ib_1} + \alpha_{jb_2} - 1\right) t_{b_1 b_2} + p_j - c_j \leq M\left(1 - x_{ijk}\right)$$

$$i, j \in \Omega^v, k \in Q^v, b_1 \in H_i^v, b_2 \in H_j^v, v = A,B \quad (23)$$

$$c_i - c_T \leq M(1 - x_{iTk}) \qquad i \in \Omega_0^v, k \in Q^v, v = A,B \quad (24)$$

$$c_i + p_j - c_j \leq 0 \qquad (i,j) \in \Phi^v, v = A,B \quad (25)$$

$$c_i + p_j - c_j \leq M\left(1 - y_{j\vartheta}\right) \qquad j \in L^v, \vartheta \in \Theta_j^v, \left(i,\vartheta\right) \in \Phi_1^v, v = A,B \quad (26)$$

$$c_j + p_i - c_i \leq M\left(1 - y_{j\vartheta}\right) \qquad j \in L^v, \vartheta \in \Theta_j^v, \left(\vartheta,i\right) \in \Phi_2^v, v = A,B \quad (27)$$

$$c_i + p_j - c_j \leq M\left(2 - y_{i\vartheta_1} - y_{j\vartheta_2}\right) \qquad i, j \in L^v, \vartheta_1 \in \Theta_i^v, \vartheta_2 \in \Theta_j^v, \left(\vartheta_1, \vartheta_2\right) \in \Phi_3^v, v = A,B \quad (28)$$

$$\sigma_{ij} = 6s^{-1}\left(\sum_{v\in V} \sum_{b\in H_j^v} b\alpha_{jb} - b(i) + l_A^F + d\left(A,B\right)\right)\hat{t} \qquad (i,j) \in \overline{\Phi} \quad (29)$$

$$c_i + \sigma_{ij} + p_j - c_j \leq 0 \qquad (i,j) \in \overline{\Phi} \quad (30)$$

$$c_i + p_j - c_j \leq M\left(1 - z_{ij}\right) \qquad i, j \in \Omega^v, v = A,B \quad (31)$$

$$c_j - p_j - c_i \leq M z_{ij} \qquad i, j \in \Omega^v, v = A,B \quad (32)$$

$$z_{ij} + z_{ji} \geq \alpha_{ib_1} + \alpha_{jb_2} - 1 \qquad i, j \in \Omega^v, b_1 \in H_i^v, b_2 \in H_j^v, \left(b_1, b_2\right) \in \Psi^v,\ v = A,B \quad (33)$$

$$\sum_{u\in\Omega_0^v} x_{uih} + \sum_{u\in\Omega_0^v} x_{ujk} + \alpha_{ib_1} + \alpha_{jb_2} \leq 3 + z_{ij} + z_{ji}$$

$$i, j \in \Omega^v, b_1 \in H_i^v, b_2 \in H_j^v, (b_1, b_2, h, k) \in \hat{\Delta}(v), v = A,B \quad (34)$$

$$c_i + \Delta_{b_1 b_2}^{hk}(v) + p_j - c_j \leq M\left(5 - \alpha_{ib_1} - \alpha_{jb_2} - z_{ij} - \sum_{u\in\Omega_0^v} x_{uih} - \sum_{u\in\Omega_0^v} x_{ujk}\right)$$

$$i, j \in \Omega^v, b_1 \in H_i^v, b_2 \in H_j^v, (b_1, b_2, h, k) \in \hat{\Delta}(v), v = A,B \quad (35)$$

$$c_j + \Delta_{b_1 b_2}^{hk}(v) + p_i - c_i \leq M\left(5 - \alpha_{ib_1} - \alpha_{jb_2} - z_{ji} - \sum_{u\in\Omega_0^v} x_{uih} - \sum_{u\in\Omega_0^v} x_{ujk}\right)$$

$$i, j \in \Omega^v, b_1 \in H_i^v, b_2 \in H_j^v, (b_1, b_2, h, k) \in \hat{\Delta}(v), v = A,B \quad (36)$$

$$c_i \leq w^v \qquad i \in \Omega^v, v = A,B \quad (37)$$

$$x_{ijk} \in \left\{0,1\right\} \qquad i, j \in \Omega^v, k \in Q^v, v = A,B \quad (38)$$

$$z_{ij} \in \left\{0,1\right\} \qquad i, j \in \Omega^v, v = A,B \quad (39)$$

$$y_{i\vartheta} \in \left\{0,1\right\} \qquad i \in L^v, \vartheta \in \Theta_i^v, v = A,B \quad (40)$$

$$\alpha_{ib} \in \left\{0,1\right\} \qquad i \in \Omega^v, H_i^v, v = A,B \quad (41)$$

$c_i \geq 0 \qquad i \in \Omega^v, v = A, B \quad (42)$

$\sigma_{ij} \geq 0 \qquad \forall (i, j) \in \overline{\Phi} \quad (43)$

In this model 0 and T are two dummy tasks, with null processing time, and $\Omega_0^v = \Omega^v \cup \{0\}$, $\Omega_T^v = \Omega^v \cup \{T\}$. The objective function (12) is a linear combination of the makespan of the two ships and the total time the containers directly transhipped spend in the crane buffers. Constraints (13) to (16) are the classical routing constraints and define the sequence of tasks performed by each crane, starting and ending with the dummy tasks 0 and *T*, respectively. We note that if 0 and *T* are the only tasks handled by a crane, then that crane is actually idle.

The stowage position, in terms of slots and bays, of the directly transhipped containers is assigned by constraint (17) and (18). The following two groups of constraints state the link between the variables y's and α's, and x's and α's. Constraints (19) ensure that a directly transhipped container is assigned to a compatible bay if and only if it is assigned to compatible slot in that bay. Note that, for each fixed $i \in L^v$, summing up constraints (19) for all the compatible bays $b \in H_i^v$, and taking into account (18), one gets $\sum_{\vartheta \in \Theta_i^v} y_{i\vartheta} = 1$, which, together with (17), guarantees the assignment of a container in L^v to one and only one slot in Θ_i^v. Constraints (20) impose that if a task is assigned to a bay then it has to be assigned to a crane able to operate that bay.

Constraints (21) to (24) compute the completion times of the tasks. In particular constraints (23) deserve to be delved. Clearly they are useful whenever the assignment $x_{ijk} = 1$ takes place. In this case we need to know the time the crane k takes to reach the bay where j is located, $b(j)$, starting from the bay $b(i)$ where i is located: $t_{b(i)b(j)}$. The latter time is a non linear function of the variables α_{ib}, α_{jb}, (whenever $i \in L^v$ or $j \in L^v, v = A, B$). In the linear constraints (23), imposed for all compatible bays H_i^v and H_j^v, the crane travel time $t_{b(i)b(j)}$ is correctly computed. Actually

$$(\alpha_{ib_1} + \alpha_{jb_2} - 1)t_{b_1b_2} = t_{b(i)b(j)} \Leftrightarrow \alpha_{ib_1} = \alpha_{jb_2} = 1 \Leftrightarrow b_1 = b(i), b_2 = b(j)$$

In the other cases ($\alpha_{ib_1} = 0$ or $\alpha_{jb_2} = 0$) constraints (23) are useless, because they are dominated by the one corresponding to $(i, j, k, b(i), b(j))$.

Constraints (25)-(28), and (30) impose the precedence relationships. In particular, (25) define the precedences between pairs of tasks whose stowage position is known, while (26) to (28) take into account possible precedence relationships involving directly transhipped containers to be loaded (see also definitions (3), (4), and (5)). Constraints (29) compute the transportation time, from the discharging to the loading bay, for each directly transhipped container: $\sigma_{ij}, (i, j) \in \overline{\Phi}$. These variables are needed in constraints (30), so as to ensure that such containers are first discharged from one vessel and then loaded on the other one, and in the objective function, for computing their waiting times in the crane buffers.

Constraints (31) and (32) define the link among z's variables and the completion times of the tasks, for all pair of tasks on the same vessel. Note that the z's variables induce a partial time-ordering on the processing of the tasks: when $z_{ij} = 1$ the task i is completed before the processing of j starts, independently from the cranes handling the two tasks. In this case z_{ji} must be necessarily equal to zero, which is ensured by constraints (31) and (32), since $z_{ji} = 1$

would result in a contradiction. However $z_{ij} = z_{ji} = 0$ is a feasible assignment for the above constraints and corresponds to a simultaneous processing of the tasks.

Relations (33) to (36) are non-simultaneity constraints. In more detail, (33) impose that the condition $z_{ij} + z_{ji} \geq 1$ holds, for each pair of tasks (i, j) in bays $(b_1, b_2) \in \Psi^v$; (34) impose the same condition for each pair of tasks (i, j) in the bays (b_1, b_2), handled, respectively, by the cranes h and k, such that $(b_1, b_2, h, k) \in \hat{\Delta}(v)$. Thus, in both cases, at least one of the variables z_{ij} and z_{ji} must be equal to 1. In light of the previous observations, constraints (31) and (32) will force either $z_{ij} = 1$ or $z_{ji} = 1$, ensuring that tasks i and j will not be processed simultaneously. Finally, constraints (35) and (36) correctly state the relation between the completion times of the tasks i and j for which (33) or (34) enforced either $z_{ij} = 1$ or $z_{ji} = 1$, taking into account the time $\Delta_{b_1 b_2}^{hk}(v)$ needed for avoiding interference between cranes. Actually, when $\alpha_{ib_1} = 1$, $\alpha_{jb_2} = 1$, $\sum_u x_{uih} = 1$, $\sum_u x_{ujk} = 1$, $(b_1, b_2, h, k) \in \hat{\Delta}(v)$, (35) and (36) become, respectively:

$$c_i + \Delta_{b_1 b_2}^{hk}(v) + p_j - c_j \leq M(1 - z_{ij})$$

$$c_j + \Delta_{b_1 b_2}^{hk}(v) + p_i - c_i \leq M(1 - z_{ji})$$

while, in the other cases, they are redundant.

Constraints (37) define the makespan of the two vessels and constraints (38)-(43) the domains of the decisional variables.

3.4 Some remarks on the DCTP model

The DCTP model described in the previous Section integrates two decisional processes: assigning a stowage position to the directly transhipped containers and determining the working schedules of the cranes operating on the two vessels. From this point of view, the DCTP generalizes the QCSP. Actually when no direct transhipment operation has to be performed, the sets L^v, D^v and those defining the precedence sets (2), (3), (4), and (5), are empty; the variables α 's, y 's, and σ 's need not to be considered, as well as the constraints involving them. In this case the resulting DCTP separates in two standard QCSPs, one for each vessel. On the other hand, assuming $\bar{\Phi} = \varnothing$ and removing the σ 's variables and constraints (29) and (30), the DCTP model allows to deal with two non-standard QCSPs, where $L^v, v = A, B$ represent sets of conventional export containers, whose stowage positions have to be decided contextually with the crane scheduling. In this scenario the ship planners could gain a wider degree of freedom in planning crane's working sequences, while reducing the service time of the vessels.

Finally we observe that the objective function is composed by three terms: the weighted makespan of the vessels and the total sojourn time of the directly transhipped containers in the crane buffers. The weight of a vessel represents the vessel priority. A suitable setting of π^A and π^B allows to model, for example, the exchange of containers between a mother vessel and a feeder.

The third term in the objective function is composed by differences of the task completion times. Therefore reducing the total sojourn time of the directly transhipped containers could be conflicting with respect to the minimization of the makespan.

Actually the DCTP is intrinsically a multi-objective optimization problem and the reduction to the scalar function (12) is done via a standard linear combination of the three objective functions. Therefore the optimal solution of the model (12)-(43) will be weakly efficient.

4 CONCLUSION

In this paper we have introduced a new operative problem arising in the management of a maritime transhipment container terminal. The direct transhipment of containers between two vessels consists of scheduling all the unloading and loading operations to be performed, taking into account that the stowage position is unknown for some containers to be loaded. The aim is to minimize the service time of both vessels, while reducing, as far as possible, the temporary storage of the directly transhipped containers. We have proposed a Mixed Integer Linear Program for the DCTP, which has been shown to be very flexible, meaning that it is able to model different scenarios.

The direct transhipment of containers, by skipping the storage phase, would allow to reduce possible yard congestions and, at the same time the storage costs. Therefore it seems to be a profitable operational modality both for the terminal management and the shipping line companies.

REFERENCES

Alicke, K. (2002), Modeling and optimization of the intermodal terminal Mega Hub. *OR Spectrum*,. 24(2), 1-18.

Bierwirth, C. and Meisel, F. (2009), A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling,* 12, 345–360.

Blumenhagen D. (1981), Containerization and Hinterland traffic. *Maritime Policy and Management*, 8(3), 197-206.

Boysen, N., Fledner, M., and Scholl, A. (2010), Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*, 32, 135-161.

Kim, K.H. and Park, Y.-M. (2004), A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156(3), 752-768.

Lee, B.K., Jung, B.J., Kim, K.H., Park, S.O. (2006), A Simulation study for designing a rail terminal in a container port. Proceedings of the *38th Conference on Winter Simulation.*

Lian, C., Hwang, H., and Gen, M. (2011), A berth allocation planning problem with direct transhipment consideration. *Journal of Intelligent Manufacturing*, DOI 10.1007/s10845-011-0566-9.

Meisel, F. (2009), *Seaside Operations Planning in Container Terminals.* Berlin: Physica-Verlag.

Moccia, L., Cordeau, J.F., Gaudioso, M., Laporte, G. (2006), A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics,* 53(1), 45-59.

Monaco, M.F. and Sammarra, M. (2011), The quay crane scheduling problem with time windows, one-way, and spatial constraints. *International Journal of Shipping and Transport Logistics*, 3(4), 454-474.

Nam, H. and Lee, T. (2012), A scheduling problem for a novel container transport system: a case of mobile harbor operation schedule. *Flexible Services and Manufacturing Journal*. DOI 10.1007/s10696-012-9135-6.

REAL-TIME CONTAINER STORAGE LOCATION ASSIGNMENT AT A SEAPORT CONTAINER TRANSSHIPMENT TERMINAL PART II

Matthew E. H. Petering

University of Wisconsin—Milwaukee, Industrial and Manufacturing Engineering Dept., P.O. Box 784, Milwaukee, WI 53201, U.S.A., mattpete@uwm.edu

ABSTRACT

This study shows how a seaport container terminal's overall productivity depends on the system that is used for automatically selecting storage locations for export containers in real time as they arrive at the terminal. This research expands upon the work that was presented at LOGMS 2010. In 2010, we defined the problem and presented the results of an initial experiment. The results illustrated the main trade-offs involved in container storage decisions and validated our main methodological approach: discrete event simulation. The current study considers three major extensions of our 2010 work. First, we do a comprehensive evaluation of the relative importance of four different objectives for making container storage decisions. This evaluation is surprisingly rare in that we are essentially testing a real-world decision making system being sold to transportation and logistics companies—in this case the Expert Decking system sold by Navis—in a controlled academic environment. Second, we investigate the role of math programming in making container storage decisions. Here, we use a simulation model to directly compare decision-making algorithms that are strictly rule-based to those that have a math programming component. Third, we perform the first academic study of a DOS (duration of stay)-based storage system for export containers at a container terminal and, to our knowledge, the most comprehensive academic study of a real-time DOS-based storage system in any setting. The DOS storage concept was introduced by Goetschalckx and Ratliff in a Management Science article in 1990. Our experimentation considers a multiple-berth, vessel-to-vessel transshipment terminal and the container storage system's interaction with other decision making systems.

Keywords: container storage location assignment, real-time control, Navis™ SPARCS Expert Decking, duration-of-stay-based storage

1 INTRODUCTION

Today, almost all overseas shipping of furniture, toys, footwear, clothing, auto parts, and electronics components is done via standardized 20', 40', and 45' long steel containers aboard deep-sea container vessels. In addition, the amount of fruit, vegetables, fish, meat, and general foodstuffs shipped in refrigerated containers is increasing. With today's just-in-time global supply chain, improving the efficiency of container shipping processes is more important than ever. This study focuses on operational control problems at seaport container terminals. Container terminals are the places in seaports where container vessels are loaded and unloaded, and where containers are temporarily stored while awaiting a future journey.

This study shows how a seaport container terminal's overall productivity depends on the system that is used for automatically selecting storage locations for export containers in real time as they arrive at the terminal. This research expands upon the work that was presented at LOGMS 2010. In 2010, we defined the problem and presented the results of an initial experiment. The results illustrated the main trade-offs involved in container storage decisions and validated our main methodological approach: discrete event simulation.

The current study considers three major extensions of our 2010 work. First, we do a comprehensive evaluation of the relative importance of the four objectives for making container storage decisions that are listed in Section 2. This evaluation is surprisingly rare in that we are essentially testing a real-world decision making system being sold to transportation and logistics companies—in this case the Expert Decking system sold by Navis (Navis Expert Decking, 2011)—in a controlled academic environment. Second, we investigate the role of math programming in making container storage decisions. Here, we use a simulation model to directly compare decision-making algorithms that are strictly rule-based to those that have a math programming component. Third, we perform the first academic study of a DOS (duration of stay)-based storage system for export containers at a container terminal and, to our knowledge, the most comprehensive academic study of a real-time DOS-based storage system in any setting. The DOS storage concept was introduced in 1990 (Goetschalckx and Ratliff, 1990). Our experimentation considers a multiple-berth, vessel-to-vessel transshipment facility and the container storage system's interaction with other decision making systems.

Although the immediate context of this work is container shipping, this study touches upon several topics of general importance to the operations research field: (1) real-time decision making in a stochastic environment; (2) the embedding of math programming routines within discrete event simulation models; (3) the relationship between the variability of a real-world system and variability within a mathematical and/or computer model for making decisions regarding the system; (4) design and management of facility operating systems and facility control systems.

2 PROBLEM DESCRIPTION

In this study, we investigate how various real-time container storage location assignment algorithms affect the overall, long-run performance of a seaport container terminal as measured in terms of GCR (gross crane rate, quay crane rate). GCR is defined as the average number of lifts achieved at a terminal per quay crane (QC) working hour and is probably the most important measure of operational performance at a container terminal. QCs are the machines that load and unload vessels. To attain a high GCR, the flow of containers back and forth between the shore and the yard has to proceed smoothly, so that QCs do not incur idle time waiting for the vehicles (yard trucks, YTs) that move cargo between the quay and storage yard. The YTs, in turn, would prefer not to wait to be served by the yard cranes (YCs) that transfer containers between YTs and the stacks in the storage yard.

In 2010, we presented four ways for the storage system to maximize GCR:

A) minimize container travel distance from quay to yard during unloading/storage
B) minimize container travel distance from yard to quay during retrieval/loading
C) minimize storage yard congestion near cargo storage locations during unloading/storage
D) minimize storage yard congestion near cargo storage locations during retrieval/loading.

The current study has two immediate goals: (1) to evaluate the relative importance of the four objectives shown above in various terminal environments and (2) to identify specific real-time container storage assignment systems that maximize GCR at one or more terminals

3 LITERATURE REVIEW

The literature relevant to this study includes all papers on container terminals that discuss (1) container storage location assignment, (2) simulation modeling, or (3) the literature itself. A total of 70 such papers were found by the author. Excellent surveys of recent research on container terminal operations have been conducted by Stahlbock and Voss (2008), Steenken *et al* (2004), and Vis *et al* (2003). A good description of container terminal operations is given in Günther and Kim (2006). A concise summary of the various operational decisions made in container terminals is given in Murty *et al* (2005).

The problem of container storage location assignment is addressed by only a handful of articles in the literature. Of particular interest here is the storage of export containers. Out of 22 such papers, only Dekker *et al* (2006) and Borgman *et al* (2010) present models in which the arrival, stay, and departure of each container are explicitly modeled. In addition, Dekker *et al* (2006) and Borgman *et al* (2010) are the only such articles that present a model that obtains numerical results on real-time container storage location assignment. Moreover, no article presents a model that shows how alternate real-time container storage location assignment systems affect the overall, long-run performance (e.g. GCR) of a multiple-berth container terminal. This research, on the other hand, has produced such a model—a discrete event simulation model that has been used to study several container terminal problems (Petering 2009, Petering 2010, Petering 2011).

4 CONTAINER STORAGE LOCATION ASSIGNMENT SYSTEM

The real-time container storage location assignment system developed in this research is the result of several ideas that have evolved over the last five years. The core of this system is a seven-step algorithm that uses rule-based and math programming features to decide the storage location for each container. This system is embedded within a simulation model.

5 DISCRETE EVENT SIMULATION MODEL

The simulation model is intended to provide a partial answer to the recent call for a more integrated approach to research on container terminal management (Stahlbock and Voss, 2008). The model has been developed based on the author's observations and extended discussions with managers at several container terminals. It simulates the activities associated with individual containers, vessels, QCs, YCs, YTs, and stacks in the yard over an arbitrarily long, user-defined time period. There are 28 types of events that can occur to affect the state of the terminal and its future evolution. The model's quick runtime has allowed us to simulate roughly 1.3 billion QC lifts worth of activity—almost three times the annual amount of activity at all of the world's container ports combined—in the following three experiments put together.

6 EXPERIMENT ONE: EXPERT DECKING

The main results of this experiment are as follows. First, the overall productivity at a terminal where an Expert Decking system is used (i.e. in which objectives A-D are intelligently incorporated into the real-time container storage location assignment system) is 1-7% higher than the productivity at a terminal in which containers are stored randomly. Second, at small- and medium-sized terminals (1-6 berths), it is more important to pursue objectives C and D—i.e. to choose storage locations that minimize the expected congestion associated with the storage and retrieval of containers in the yard—than objectives A and B—i.e. to choose locations that minimize the expected distance containers travel to/from the yard.

7 EXPERIMENT TWO: MATH PROGRAMMING APPROACH

This experiment considers if a math programming approach can improve the performance of an otherwise rule-based container storage system. As many people have observed, container terminal operations exhibit a high level of unpredictability and uncertainty. For example, the time taken by a QC (YC) to handle a single container typically ranges from 1-2 (1.2-3.4) minutes. In addition, a YT's (YC's) actual travel (gantry) time for a given journey may range from 30% below to 30% above the expected time. Furthermore, when equipment is manually controlled there is no way to know the exact time when a container move or journey will be completed prior to its completion. After all, the duration of the final part of a process (e.g. the container handoff between a crane and a truck or the fine-positioning of a crane or truck at precisely the correct location) is highly unpredictable.

Unfortunately, math programming techniques are not well suited to the above real-time environment. For example, plans and schedules for detailed QC/YT/YC operations that are based on math programming models with a look-ahead horizon become out of date almost immediately after their construction. Indeed, the deterministic process durations in such models (e.g. 1.5 min per QC lift and 2.2 min per YC lift) are rarely achieved in practice. Reconstructing such plans and schedules in real time using the latest information coming out of the TOS (terminal operating system) may not be practical because runtimes for math programming heuristics may be too long to be viable in a real-time environment. For example, as many as 25,000 containers may be transferred between ships and shore during a busy day at a large terminal. For each container, a minimum of three operational decisions are made: selecting the storage/retrieval location in the yard, deciding which YC handles the container, and deciding which YT transports the container. This amounts to roughly 75,000 operational decisions per day. With only 86,400 seconds in a day, the average inter-decision time is therefore about one second. The actual inter-decision time is highly stochastic and is sometimes less than one tenth of a second. To be viable in a real-time environment, a decision making algorithm should therefore have a runtime that is less than 1 second on 100% of occasions. Otherwise, operations will literally grind to a halt as equipment operators wait in a queue for instructions from the TOS. In summary, math programming techniques do not have good potential for use in the detailed scheduling of QC/YT/YC operations at the individual container level at a terminal where equipment is manually controlled. We therefore do not expect such optimization techniques to be useful for making storage decisions at the individual container level.

A yard template, on the other hand, is a concept that can integrate math programming into container storage decisions. A yard template is a priority-based storage scheme that pre-assigns to each vessel (i.e. liner service) certain high priority storage locations where cargo loaded onto the vessel should be stored. A yard template is created offline by solving a math programming problem. It remains unchanged as long as the terminal's weekly schedule, its weekly berth plan, and the expected cargo volume loaded onto each vessel is unchanged. The creation of a yard template is a tactical decision that affects container storage decisions at an operational level.

Yard templates can be used to pursue two of the four objectives listed in Section 2: B—minimizing average container travel distance from yard to quay during loading—and D—minimizing YT congestion in the vicinity of container storage locations when containers are being retrieved. A yard template can work towards objective B by putting a vessel's high priority storage locations close to the vessel's home berth. It can work towards objective D by spreading throughout the yard the high priority storage areas for cargo loaded onto vessels that are expected to be present at the same time.

Results from the attempt to integrate yard templates that are optimized using math programming procedures into a real-time container storage system are two-fold. First, it remains to be seen whether math programming methods have a role to play in container storage location assignment. Indeed, we show that math programming-based, template-based storage systems, which assign to each vessel certain high-priority storage areas where cargo loaded onto the vessel is to be stored, perform no better than simple, rule-based alternatives. Second, the results from experiments 1-2 show that, for most terminals we consider, container storage objectives A-D can be ranked in the order C, D, A, B from most to least important.

8 EXPERIMENT THREE: DURATION-OF-STAY (DOS)-BASED STORAGE

A DOS-based storage system is similar to the turnover-based storage systems commonly used in warehouses. The idea of a DOS-based storage system is to place items with low expected lifetime in the facility (i.e. high "turnover") close to the facility's entrance/exit point in order to minimize the travel distance associated with the storage and retrieval of an average item (Goetschalckx and Ratliff, 1990). Transshipment container terminals—where all cargo enters and leaves the terminal by vessel—are unlike import/export terminals in that the entrance/exit point(s), as in many warehouses, are often only on one side of the facility—at the quay. Thus, a DOS-based storage concept might increase GCR at a transshipment container terminal.

In our implementation of a DOS-based storage concept, we consider *stacks of containers*, not containers themselves, as the "unit loads" or "items" in a warehouse. We also assume the terminal adopts a group-based *homogenous stacking policy* where all containers in a stack must belong to the same *group*. Thus our analysis should not to be confused with *container dwell-time-based storage systems* for *import* containers in which containers from different groups are mixed in the same stack and the goal is to minimize the number of instances when YCs shift containers between stacks in order to dig out individual containers at lower tiers.

Results are as follows. A DOS-based storage system is not suitable for a multiple-berth terminal where container stacks are high. Rather, it is most suitable for a narrow, deep, single-berth transshipment terminal where container stacks are low and YCs are plentiful. However, even in the most favorable circumstances, a pure DOS-based storage system yields negligible to small GCR gains versus a random storage system.

9 CONCLUSION

This study shows how a seaport container terminal's overall productivity depends on the system that is used for automatically selecting storage locations for export containers in real time as they arrive at the terminal. This research expands upon the work that was presented at LOGMS 2010. Our experimentation considers a multiple-berth, vessel-to-vessel transshipment terminal. Future effort will focus on non-transshipment terminals and terminals that use other kinds of equipment such as straddle carriers, top handlers, and reach stackers.

REFERENCES

Borgman, B., van Asperen, E., & Dekker, R. (2010). Online rules for container stacking. *OR Spectrum*, **32**, 687-716.

Dekker, R., Voogd, P., & van Asperen, E. (2006). Advanced methods for container stacking. *OR Spectrum*, **28(4)**, 563-586.

Goetschalckx, M., & Ratliff, H. D. (1990). Shared storage policies based on the duration stay of unit loads. *Management Science*, **36**, 1120-1132.

Günther, H.-O., & Kim, K.-H. (2006). Container terminals and terminal operations. *OR Spectrum*, **28(4)**, 437-445.

Murty, K. G., Liu, J., Wan, Y. W., & Linn, R. (2005). A decision support system for operations in a container terminal. *Decision Support Systems*, **39(3)**, 309-332.

Navis Expert Decking brochure (website accessed Nov. 11, 2011): http://www.navis.com/pdf/products-datasheets/ds_expert_decking.pdf.

Petering, M. E. H. (2009). Effect of block width and storage yard layout on marine container terminal performance. *Transportation Research E*, **45(4)**, 591-610.

Petering, M. E. H. (2010). Development and simulation analysis of real-time, dual-load yard truck control systems for seaport container transshipment terminals. *OR Spectrum*, **32**, 633-661.

Petering, M. E. H. (2011). Decision support for yard capacity, fleet composition, truck substitutability, and scalability issues at seaport container terminals. *Transportation Research E*, **47(1)**, 85-103.

Stahlbock, R., & Voss, S. (2008). Operations research at container terminals: a literature update. *OR Spectrum*, **30(1)**, 1-52.

Steenken, D., Voss, S., & Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, **26(1)**, 3-49.

Vis, I. F. A., & de Koster, R. (2003). Transshipment of containers at a container terminal: an overview. *European Journal of Operational Research*, **147(1)**, 1-16.

IDENTIFICATION OF LATENESS FACTORS IN CONTAINER HANDLING PROCESS USING BAYESIAN NETWORK FROM EVENT LOG

Riska Asriana Sutrisnowati[1], Daeuk Jeon[2], Hyerim Bae[2], Minseok Song[3], Hoon Lee[4] and Joonsoo Bae[5]

[1]*Institute of Logistics Information Technology, Pusan National University, South Korea, riska@pusan.ac.kr*

[2]*Department of Industrial Engineering, Pusan National University, South Korea,*

*{jdw, *hrbae}@pusan.ac.kr*

[3]*School of Technology Management, Ulsan National Institute of Science and Technology, South Korea, msong@unist.ac.kr*

[4]*Logistics Institute, Total Soft Bank Ltd., South Korea, pisycho@tsb.co.kr*

[5]*Department of Industrial and Information Systems Engineering, Chonbuk National University, South Korea, jsbae@chonbuk.ac.kr*

ABSTRACT

The handling of containers in port logistics can be classified into different business process types, such as discharging, loading, gate-in, and gate-out each of which includes related, structured activities that are carried out using various equipments, including quay cranes, and yard crane, among others. The high interdependency of processes and equipments on various factors, schedules can be missed in real time, resulting in undesirable process down time and lateness. So that if there is one late container process can negatively affect the scheduling of successive processes. Identifying the causes of lateness, therefore, is a necessity. The specific purpose of the present study was to identify the most significant causes of lateness by considering various factors related cargo types, workers, and others. First, we examined the event logs obtained from port information systems and discovered process models from an open source process miner, ProM, using Heuristic Miner. Next, we derived a method for generation of Bayesian Network from the process models, and also applied additional variables to that Network. Finally, we could identify the most significant causes of lateness. These, significantly, can be used to reengineer existing processes for better performance.

Keywords: Bayesian Network, Process Mining, Port Logistics Process, Container Workflow

1 INTRODUCTION

Container operations in the field of port logistics involve various processes including loading, discharging, special operations, and others. These processes are often conducted in parallel and planned schedule. The complexity and high interdependency of container operations make container operation scheduling an NP-hard problem (Mak & Zhang, 2009). Since container operations are inter-related, and sometimes cyclical, late handling of a container can affect succeeding operations, thereby the entire operation becomes off schedule.

Nowadays, most port logistics include supporting information systems that enable retrieval of information properties in the form of event logs. Using process mining, a process model can be discovered from the event logs. From the process model, the precedence relation among pertinent events and actual container operations can be obtained. The process model also can be analysed to determine its conformance checking (Rozinat & Aalst, 2008) and predict the completion time, sojourn, and elapsed time (Aalst, Schonenberg, & Song, 2011). However, this process model cannot use to identify the factors causing lateness. The present study employs the Bayesian Network to investigate the factors contributing to lateness in container operation.

The remainder of this paper is organized as follows. Section 2 discusses backgrounds, section 3 presents the Bayesian Network construction, and section 4 draws conclusions.

2 BACKGROUNDS

2.1 Bayesian Network

A Bayesian Belief Network or Bayesian Network, as based on the work of Mitchell (Mitchell, 1997), provides a joint probability distribution specifying a set of conditional independence assumptions and represented by a directed acyclic graph (DAG) together with sets of local conditional probabilities. Each variable in the joint space is represented by a node in the Bayesian Network and each links between two nodes probabilistically dependence.

The joint probability for any desired assignment of values $\langle y_1, \ldots, y_n \rangle$ to the tuple of network variables $\langle Y_1, \ldots, Y_n \rangle$ can be computed by the formula

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i)) \quad (1)$$

where $Parents(Y_i)$ denotes the set of immediate predecessors of Y_i in the network. The values of $P(y_i | Parents(Y_i))$ are stored in the conditional probability table associated with node Y_i.

2.2 Graph Decomposition

Aho et al. (Aho, Lam, Sethi, & Ullman, 2007) defined a dependency graph as the flow of information among the attribute instances in a particular parse tree; an edge from one attribute instance to another means that the value of the first is needed to compute the second. Edges express constraints that are implied by semantic rules. According to this definition, a dependency graph without cycles or loops has a structure similar to that of a DAG.

Learning a Bayesian Network when its structure is not known in advance is difficult (Mitchell, 1997). Wibisono, in (Wibisono, 2011), has developed the knowledge discovery in Business Process Management Systems using Bayesian Network. Wibisono employed the K2 algorithm to learn the Bayesian Network structure from event data. Although this method may applicable in this study, however due to the size of event logs, this method will be a time consuming job to retrieve the Bayesian Network structure.

To obtain the corresponding Bayesian Network structure, in the present study, we employ a dependency graph and decomposed it into DAG. In an example of Figure 1, a dependency graph retrieved from the ProM Framework (Process Mining Group, 2009) using Heuristic Miner (Weijters, Aalts, & Medeiros, 2006) is presented. In this graph, the numbers in the boxes represent the activity frequencies, the numbers on the arcs indicate the reliabilities of respective causal relations, and the numbers on the nodes are the frequencies. For example, from Figure 1, the *Start* activity occurs 3705 times which 2988 instances will follow *Lift off Crane* activity with reliability of 0.996.

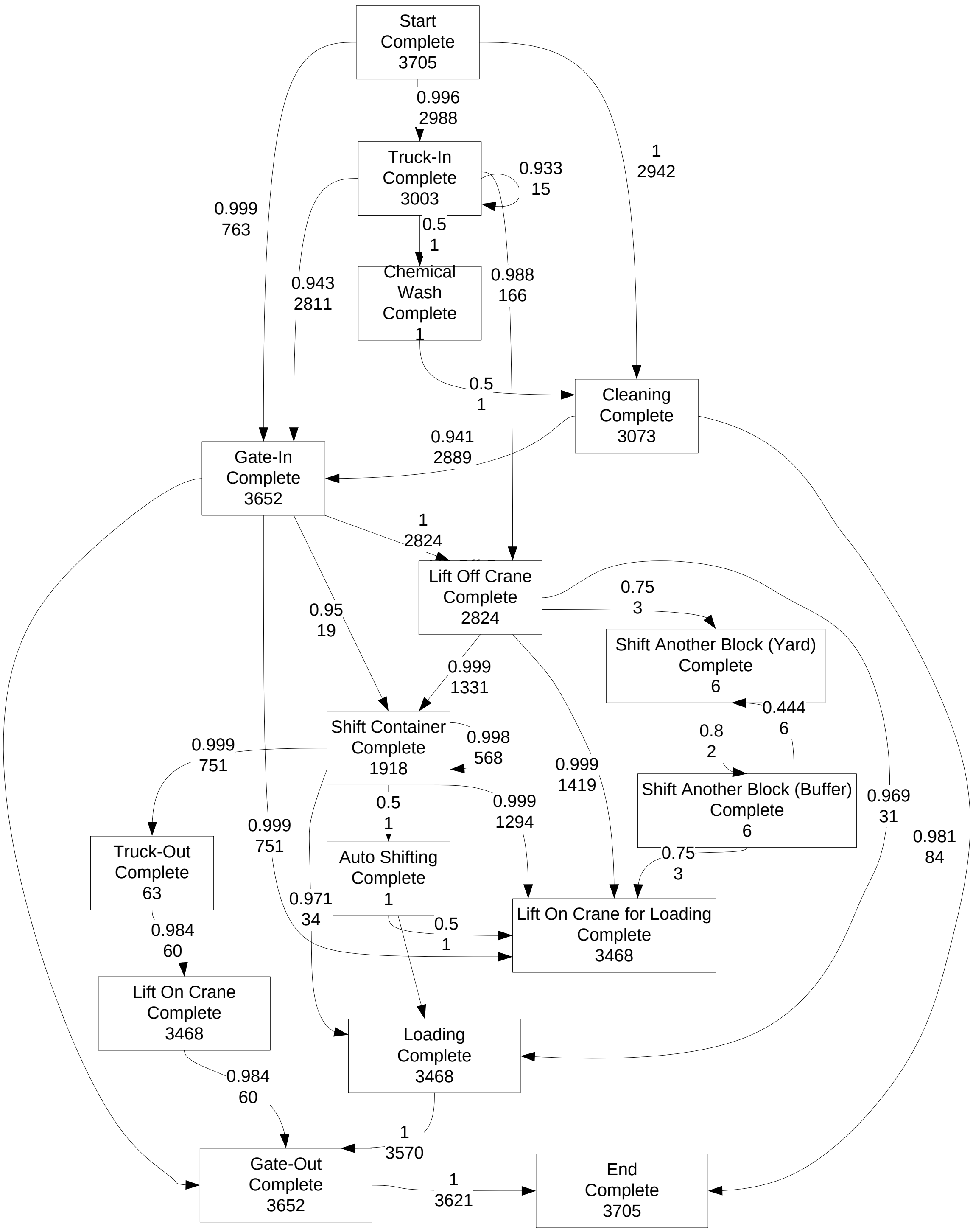


Figure 1. Dependency graph of port operation

Researchers have developed a decomposition method for a control flow graph into a tree. Bae et al. developed a decomposition of process model into a tree using ECA rules (Bae, Bae, Kang, & Kim, 2004), which originally used in active database systems. Their approach detects, from a process model, a block of processes consisting of iterative block, serial block and parallel block. However, the approach is considered a block process, it was considered to be insufficient for the purposes of the present study. Ehuis et al. developed a recursive algorithm for decomposition a dependency graph into a tree (Ehuis & Grefen, 2009). As the original purpose of decomposition was to construct a BPEL structure from an available dependency graph, and since the Bayesian Network is constructed using a DAG, we derived a method that decomposes a dependency graph into a DAG.

2.3 Total Lateness Values

Wibisono (Wibisono, 2011) defined lateness as the summation of the total differences between actual completion time, desired completion time and lateness tolerance. A positive value of lateness signifies *Late* and a negative value of *Late* means earliness or *Not Late*, formulated as follows

$$Late(a_j, \ldots, a_{j'}) = \sum_{i=j}^{j'} (\widehat{c_i} - c_i - \delta_i)e_i \,; where\ j' > j \qquad (2).$$

The function $Late(a_j, a_{j'})$ is the total lateness occurred between activity a_j and activity $a_{j'}$; $\widehat{c_i}$ is the actual completion time of an instance of the *i*-th activity while c_i is the expected completion time of the *i*-th activity, and δ_i denotes the lower bound of lateness or lateness tolerance of the *i*-th activity. If $\widehat{c_i} - c_i \geq \delta_i$, the *i*-th activity is *Late* since its value is greater than the lower bound of the time constraint. The additional variable e_i is given the value 1 if a process instance passes through activity *i*; otherwise 0.

3 BAYESIAN NETWORK CONSTRUCTION AND LATENESS ANALYSIS

This section will discuss the implementation of our approach, which consists of three steps: 1) dependency graph retrieval from the ProM framework (Process Mining Group, 2009) using Heuristic Miner (Weijters, Aalts, & Medeiros, 2006); 2) dependency graph decomposition into a DAG preparatory to construction of the Bayesian Network; 3) sensitivity analysis.

3.1 Dependency Graph Retrieval

We use ProM API to retrieve the dependency and causality matrix from Heuristic Miner (Weijters, Aalts, & Medeiros, 2006). Heuristic Miner is implemented to retrieve the dependency graph using the documentation at www.processmining.org (Process Mining Group, 2009).

3.2 Bayesian Network Construction

To construct a Bayesian Network, we need to decompose a dependency graph into DAG. The decomposition method consists of four steps: 1) initialization steps to construct the reverse-post-order and dominator tree from the graph, 2) pre-processing steps to detect the back-edges or cycle, 3) processing steps to decompose the loops into non-loop structures, and 4) finalization steps to calculate the initial probability for each node in the graph and building the Bayesian Network. This study considered only one type cargo container, which is Bundle container (BN). An algorithm detects a cycle, or in other words, a self-loop and length-one-loop. However, before we can detect a cycle, for each node we need to calculate the pre-node and post-node, the reverse post-order and the dominator tree of the graph.

STEP 1. INITIALIZATION: the construction of the reverse-post-order and dominator tree.

Ehuis and Grefen (Ehuis & Grefen, 2009), stated that, given a node n, its direct input and output set, written as pre-node(n) and post-node(n), respectively, is:

$$pre - node(n) = \{x | (x, y) \in E \wedge y = n\} \qquad (3).$$

$$post - node(n) = \{y | (x, y) \in E \wedge x = n\} \qquad (4).$$

The causality matrix from Heuristic Miner results already contains a pre-node and post-node set, but has implicit semantic for representing the XOR and AND relations. To calculate the pre-node and post-node therefore, we use the causality matrix but ignore its implicit semantic. For example, based on Figure 1, *Start* node has output *[[Cleaning], [Gate-In, Truck-In]]*, which means that it is followed by node *((Cleaning) ∨ (Gate-In ∧ Truck-In))*. In the post-node structure, the post-node of *Start* node will be [*Cleaning, Gate-In, Truck-In*].

```
PROCEDURE 1 REVERSEPOSTORDER(n, PRE, POST)
    Input: N is a set of  node {n0,...ni};
        PRE(n,k) is a set of pre-node of node {n0,...ni} which has {k0, ..., kl} pre-nodes
        POST(n,k) is a set of post-node of node {n0,...ni} which has {k0, ..., kj} pre-nodes
        PRE(n) denotes all pre-nodes of node n = {k0, ..., kl}
        POST(n) denotes all post-nodes of node n = {k0, ..., kj}
    Output: List of reverse post order RPO
    RPO ⟵{∅}
    if RPO = {∅} then
        RPO.add (POST(n))
    end if
    if POST(n).Contains (-1) then
        RPO.add(n, RPO.last())
    else
        for all i ϵ RPO do
            for all j ϵ POST(RPO(i)) do
                if POST(RPO(i),j) ≠ {∅} and POST(RPO(i),j) ∉ RPO then
                    RPO.add (REVERSEPOSTORDER(POST(RPO(i),j),PRE, POST))
                end if
            end for
        end for
    end if
    return RPO
```

Figure 2. Algorithm for construction of reverse post-order

Aho et al. (Aho, Lam, Sethi, & Ullman, 2007), defined reverse post-order as the reverse of post-ordering, that is, as a list of the vertices in the opposite order of their last visit. To find the reverse post-order of the dependency graph, we provide the recursive algorithm shown in Figure 2.

Dominator tree (Aho, Lam, Sethi, & Ullman, 2007), is defined as node *d* of a flow graph dominates node *n*, written *d* dom *n*, if every path from the entry node of the flow graph to *n* goes through *d*. Note that under this definition, every node dominates itself.

The iterative algorithm for construction of a dominator tree was firstly developed by Cooper et al. (Cooper, Harvey, & Kennedy, 2001). However, it was not designed to find a dominator tree in the case of a graph has a self-loop. For the present purposes then, we slightly modified the algorithm by adding a self-loop filter.

STEP 2. PRE-PROCESSING: the detection of the back-edges and loops.

Aho et al. (Aho, Lam, Sethi, & Ullman, 2007) stated that a back edge is an edge *a* -> *b* whose head *b* dominates its tail *a*. For any flow graph, every back edge is retreating, but not every retreating edge is a back edge.

There are several types of back edges and loops: self-loop, length-one-loop, length-two-loop, and others. We here consider only two types, self-loop and length-one-loop. As shown in Figure 1, the activity *Cleaning* is one of the examples of a self-loop, which has an edge such that its origin and destination is itself. Also shown is an example of a length-one-loop: activity *Shift to other block (Yard)* to activity *Shift to other block (buffer)*. To identify these loops, we developed the algorithm shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**, lines 7-8 of which can be used to identify the length-one-loop and lines 9-10, the self-loop.

PROCEDURE 2 CYCLEDETECTION(DT, RPO, PRE, POST)

```
Input: DT(n) is a List of dominator tree of node {n0,...ni};
       RPO(n) is a List of reverse post order of node {n0,...ni};
       PRE(n,k) is a set of pre-node of node {n0,...ni} which has {k0, ..., kl} pre-nodes
       POST(n,k) is a set of post-node of node {n0,...ni} which has {k0, ..., kj} pre-nodes
       PRE(n) denotes all pre-nodes of node n = {k0, ..., kl}
       POST(n) denotes all post-nodes of node n = {k0, ..., kj}
Output: List of cycle node CYCLE(k,l)
CYCLE ⟵{∅}
k⟵0
for all i ϵ RPO decrement do
    for all j ϵ RPO decrement do
        if i < j and DT(RPO(i)) = RPO(j) and POST(i).Contains(RPO(j)) then
            CYCLE.get(k).add(RPO(i))
            CYCLE.get(k).add(RPO(j))
            k⟵k+1
        else if i = j and POST(RPO(i)).Contains(RPO(i)) and PRE(RPO(i)).Contains(RPO(i)) then
            CYCLE.get(k).add(RPO(i))
            CYCLE.get(k).add(RPO(j))
            k⟵k+1
        end if
    end for
end for
return CYCLE
```

Figure 3. Algorithm for cycle detection

STEP 3. PROCESSING: the decomposition of the loops structure into non-loops structure.

The decomposition step initiates after the cycle detection step is completed. As mentioned above, the structure of a dependency graph without loops can be considered to be that of a DAG; therefore, in the decomposition step we simply decompose the loop structure. Figure 4 shows the decomposition structure for a self-loop, Figure 4 (a), and a length-one-loop, Figure 4 (b). The decomposition is conducted based on the Bayesian theorem (Mitchell, 1997), as follows

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (5).$$

According to Figure 4 (b), assume that the edges from node 2 to node 1 is a back edge, then we make a dummy node as a copy of node 2, and make an arc from node *dummy* to node 1, to represent the arc between node 2 and node 1.

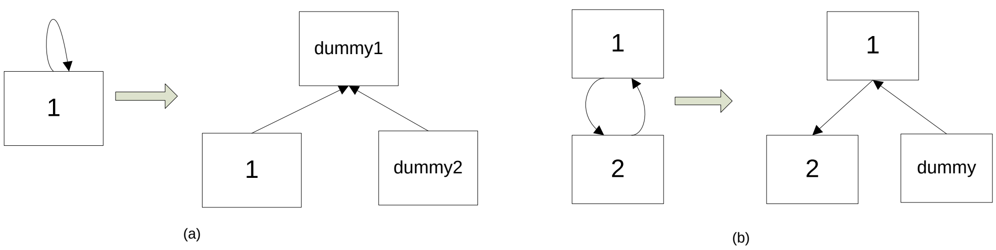


Figure 4. Decomposition of loop structure: (a) self-loop and (b) length-one-loop

We provide the algorithm for loop decomposition shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**. Lines 4-20 are the decomposition steps for a self-loop, illustrated in Figure 4 (a), and lines 21-27 are the decomposition steps for a length-one-loop, illustrated in Figure 4(b).

Procedure 3 DECOMPOSITION(*CYCLE, PRE, POST, AU, DAF*)

```
Input: PRE(n,k) is a set of pre-node of node {n_0,...n_i} which has {k_0, ..., k_l} pre-nodes
       POST(n,k) is a set of post-node of node {n_0,...n_i} which has {k_0, ..., k_j} pre-nodes
       PRE(n) denotes all pre-nodes of node n = {k_0, ..., k_l}
       POST(n) denotes all post-nodes of node n = {k_0, ..., k_j}
       CYCLE(i,j) denotes the set of cycle nodes
       AU(i,j) is a number of occurrences that goes from node i to node j
       DAF(n) is the total number of event occurrences in node {n_0,...n_i}
       temp is a List of temporary pre-nodes and post-nodes
Output: Updated PRE, POST, AU, and DAF
for all i ϵ CYCLE do
    temp ⟵ {Ø}
        if CYCLE(i,0) == CYCLE(i,1) then /*self-loop*/
            PRE.add(D1) /*Add two dummy nodes, denoted as D1 and D2*/
            PRE.add(D2)
            POST.add(D1)
            POST.add(D2)
            DAF(D1) ⟵DAF(CYCLE(i,0))/*update actual firing*/
            DAF(D2) ⟵AU(CYCLE(i,0), CYCLE(i,0))
            DAF(CYCLE(i,0)) ⟵ AU(CYCLE(i,0), CYCLE(i,0))
            temp ⟵PRE.get(CYCLE(i,0)) /*Remove all pre-nodes of Node(i,j) and make it pre-nodes of D1*/
            PRE(D1).add(temp)
            PRE(CYCLE(i,0)).remove()
            temp ⟵POST.get(CYCLE(i,0)) /*Remove all post-nodes of Node(i,j) and make it post-nodes of D1*/
            POST(D1).add(temp)
            POST(CYCLE(i,0)).remove()
            AU(D1,D2) ⟵ AU(CYCLE(i,0), CYCLE(i,0)) /*Add edges between Node(i,j) and D1, and between D2 and D1*/
            AU(CYCLE(i,0),D2) ⟵AU(CYCLE(i,0), CYCLE(i,0))
            AU(CYCLE(i,0), CYCLE(i,0)) ⟵0
        else /*length-one-loop*/
            PRE.add(D3) //Add one dummy nodes, denoted as D3
            POST.add(D3)
            DAF(D3) ⟵AU(CYCLE(i,0), CYCLE(i,1))
            AU(CYCLE(i,0),D3) ⟵ 0 /*Add edges between Node(i,j) and D1, and between D2 and D1*/
            AU(D3, CYCLE(i,0)) ⟵AU(CYCLE(i,0), CYCLE(i,1))
            AU(CYCLE(i,0), CYCLE(i,1)) ⟵0
        end if
    end for
end for
```

Figure 5. Algorithm for loop decomposition

STEP 4. FINALIZATION: the calculation of initial probability and build the network.

To construct the Bayesian Network, we need to calculate the initial probability of each node, for which we employ the arc usage and duplicate actual firing derived from Heuristic Miner graph (Weijters, Aalts, & Medeiros, 2006). Arc usage, denoted as *AU(i,j)*, is the number of occurrences that goes from node *i* to node *j*, while duplicate actual firing, denoted as *DAF(i)*, is the total number of event occurrences in node *i*. We can use the following formula to calculate the initial probability.

$$Pr(X = StartNode) = \begin{cases} \#observedoccurrence / DAF(StartNode), & if\, the\ observed\ Factor\ goes\ to\ StartNode \\ 1 - \#observedoccurrence / DAF(StartNode), & otherwise \end{cases} \quad (5).$$

$$Pr(X = EndNode) = \begin{cases} \sum_{i \,\epsilon\, pre-node(EndNode)} \dfrac{\#lateoccurrence * AU(i, EndNode) * AU(i, EndNode)}{DAF(i) * DAF(i) * DAF(EndNode)}, if\ node\ i\ is\ Late \\ 1 - \sum_{i \,\epsilon\, pre-node(EndNode)} \dfrac{(DAF(i) - \#lateoccurrence) * AU(i, EndNode) * AU(i, EndNode)}{DAF(i) * DAF(i) * DAF(EndNode)}, otherwise \end{cases} \quad (6).$$

$$Pr(X = j, j \notin \{StartNode, EndNode\}) \begin{cases} \sum_{i \,\epsilon\, pre-node(j)} \dfrac{AU(j,i) * AU(j,i)}{DAF(j) * DAF(i)}, if\ Node\ i\ goes\ to\ Node\ j \\ 1 - \sum_{i \,\epsilon\, pre-node(j)} \dfrac{(DAF(j) - AU(j,i)) * AU(j,i)}{DAF(j) * DAF(i)}, otherwise \end{cases} \quad (7).$$

3.3 Inference and Sensitivity Analysis

We used NETICA API (Norsys Software Corp.) to build a Bayesian Network and perform a sensitivity analysis. Figure 6 shows the result for the Figure 1 Bayesian Network following completion of graph decomposition into a DAG.

By using the in-built NETICA “Sensitivity to findings” function, the sensitivities can be expressed in terms of Entropy reduction, a concept borrowed from Information Theory (Han, 2005). Figure 7 shows the highest five contributing factors to the total lateness value. Among them, two factors both valued at 0.48%, node *D1_ShiftContainer* and node *Auto_Shifting*, were related to the rehandling of containers.

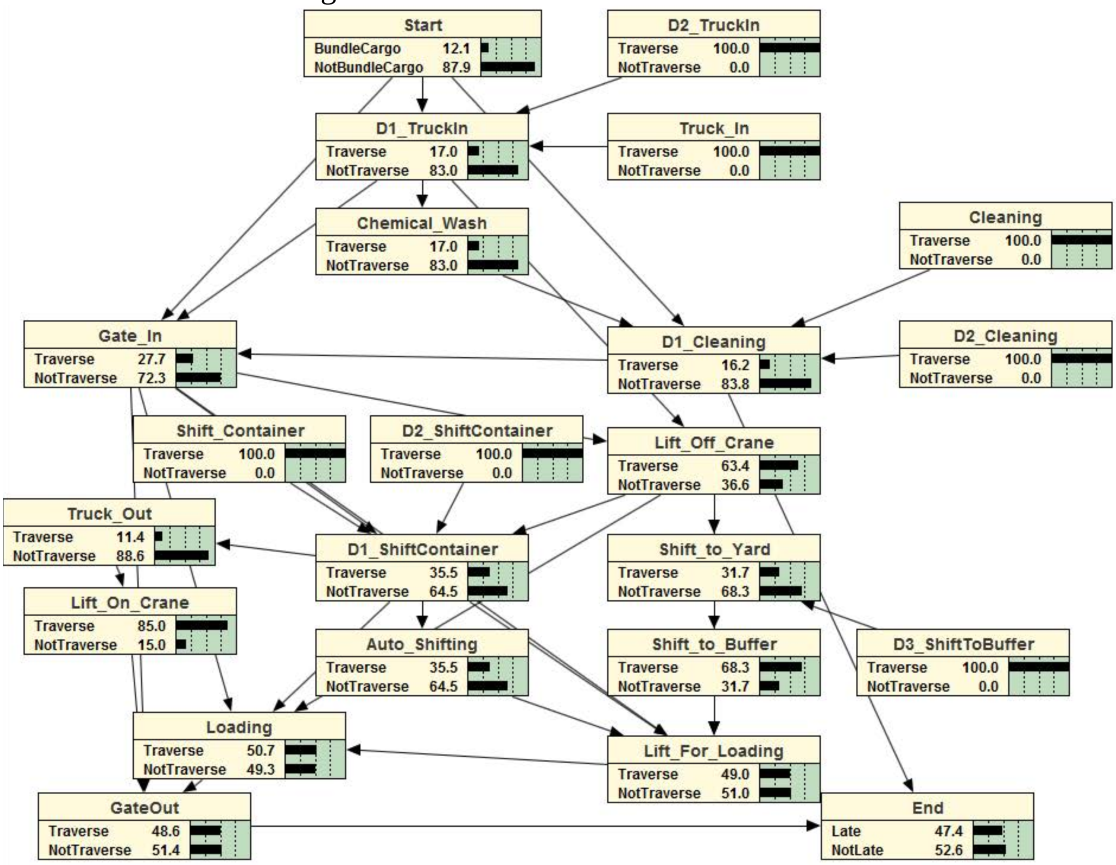


Figure 6. The result of Bayesian Network from Figure 1

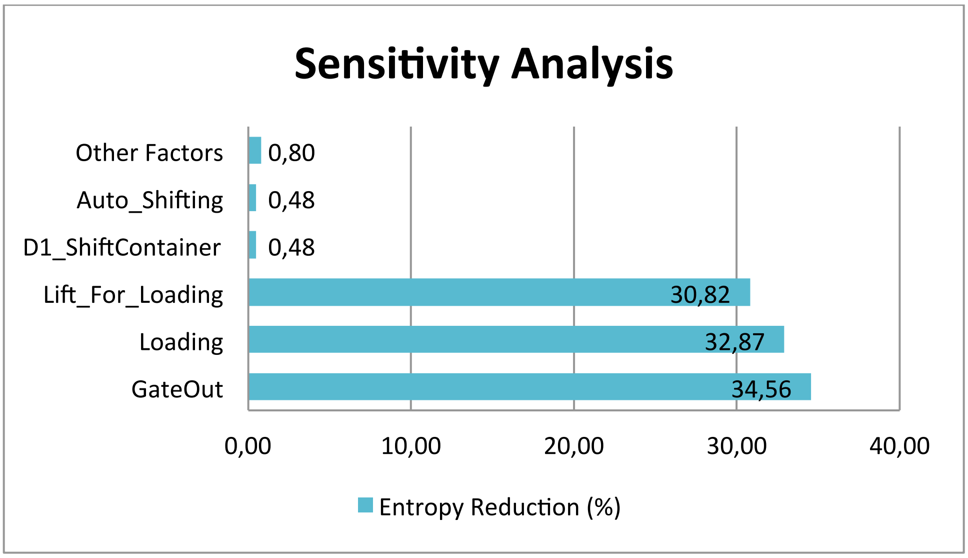


Figure 7. Sensitivity analysis result for Figure 6 Bayesian Network from

4 CONCLUSION

In this paper, we addressed the significant contributing factors to the port operation lateness. To that end, we utilized the Bayesian Network to perform a sensitivity analysis. Specifically, we employed a dependency graph results of Heuristic Miner to construct the Bayesian Network through decomposition steps that decompose a dependency graph into a directed acyclic graph (DAG). The steps consist of 1) constructing the reverse-post-order and dominator tree from the graph, 2) detecting the back-edges or cycle (in this study, we detected the self-loop and length-one-loop), 3) decomposing the loops into non-loop structures, 4) calculating the initial probability for each node in the graph and building the Bayesian Network.

The result of the Bayesian Network sensitivity analysis showed that the factors contributing to the total lateness value in the port operations, as ranked from highest to lowest, are *GateOut*, *Loading*, *Lift_For_Loading*, *D1_ShiftContainer* and *Auto_Shifting* respectively.

As noted above, this study considered only two kinds of loops: the self-loop and the length-one-loop. Long-loops detecting and predicting total lateness values will be topics for further research.

ACKNOWLEDGEMENTS

This research was supported by the MKE (The Ministry of Knowledge Economy of Korea) under the supporting business of new markets creation research program supervised by the NIPA (National IT Industry Promotion Agency). The last author is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0025650).

REFERENCES

Norsys Software Corp. (n.d.). Retrieved April 2012, from Norsys Software Corporation: http://www.norsys.com/

Aalst, W. M., Schonenberg, M., & Song, M. (2011). Time prediction based on process mining. *Information Systems* , *36* (2), 450-475.

Aho, A., Lam, M., Sethi, R., & Ullman, J. (2007). *Compilers principles, techniques, & tools* (Second Edition ed.). Pearson Education, Inc.

Bae, J., Bae, H., Kang, S.-H., & Kim, Y. (2004). Automatic control of workflow processes using ECA rules. *IEEE Transaction on Knowledge and Data Engineering , 16* (8), 1010-1023.

Cooper, K. D., Harvey, T. J., & Kennedy, K. (2001). A simple, fast dominance algorithm. *Software Practice and Experience , 4*, 1-10.

Ehuis, R., & Grefen, P. (2009). Composing service into structured process. *International Journal of Cooperative Information System , 18* (2), 309-337.

Han, J. (2005). *Data Mining: Concepts and Techniques.* San Francisco, CA, USA.: Morgan Kaufmann Publisher Inc.

Mak, K., & Zhang, L. (2009). Simultaneous scheduling of import and export containers handling in container terminals. *Proceedings of the World Congress on Engineering 2009, WCE 2009, 1.* London, U.K.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Science/Engineering/Math.

Process Mining Group, M. d. (2009). *ProM*. Retrieved May 14, 2012, from Process mining research tools application: http://www.processmining.org/prom/start

Rozinat, A., & Aalst, W. M. (2008). Conformance checking of processes based on monitoring real behavior. *Information Sytem , 33* (1), 64-95.

Weijters, A. J., Aalts, W. M., & Medeiros, A. K. (2006). Process mining with the heuristicsminer algorithm. *Technology , 166*, 1-34.

Wibisono, A. (2011). *Knowledge discovery in business process management systems using bayesian network.* Master Thesis, The Graduate School Pusan National University, Department of Logistic Information Technology.

CONSTRAINT PROGRAMMING APPROACH TO QUAY CRANE SCHEDULING PROBLEM

C. Ozgur Unsal[1] and Ceyda Oguz[1]

[1]*College of Engineering, Koc University, Sariyer, Istanbul, Turkey, cunsal@ku.edu.tr, coguz@ku.edu.tr*

ABSTRACT

This study examines quay crane scheduling problem (QCSP) in container terminals. QCSP requires completion of all loading and unloading operations of a berthed vessel under various constraints. A constraint programming (CP) model, which consists of global constraints and propositional logic, is constructed by taking numerous properties of the problem such assafety margins, travel times andprecedence relations into account. The performance of the proposed CP model is compared with some powerful algorithmspresented in recent QCSP literature. The result from the computational experiments indicates that the proposed CP model is able to produce good results for the QCSPwhile reducing the computational time.

Keywords: Quay Crane Scheduling, Constraint Programming, Container Terminals

1 INTRODUCTION

As the distance between manufacturing and consumption locations increases, the container traffic and the competition between the container terminals grow accordingly. To exist in this competitive market, the container terminal operations must be efficient. Some important seaside operations in container terminals and their relationships are shown in Fig. 1.Comprehensive surveys for the problems related to the container terminals are provided by Vis and de Koster (2003), Steenken *et al* (2004), and Stahlbock and Voß (2008).

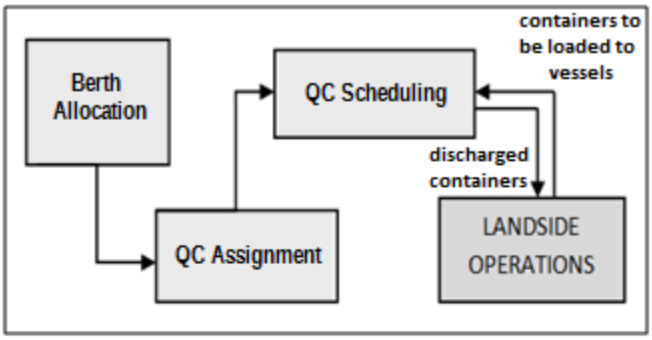


Figure 1. Seaside operations in container terminals

In this paper, scheduling of quay cranes is studied. A quay crane (QC) is huge equipment which is used in container terminals to load and unload container vessels (see Fig.2). The speed and the reliability of QC operations strongly depend on the work schedules which are projected for the QCs. In thereal-world, quay crane scheduling significantly affects the makespan of a container vessel since quay cranes are the interface between the land and the water side in any port container terminal.

The paper is organized as follows. Section 2 describes the problem and then gives a brief literature review. The methodologiesused in this study are explained in Section 3. In Section 4, a new constraint programming model is proposed for QCSP and a mixed-integer programming model is developed for a relaxed version of the QCSP to find a lower bound to the original one. Computational experiments and their results areexplained and discussed in Sections 5 and 6, respectively. The study is concludedwith Section 7.

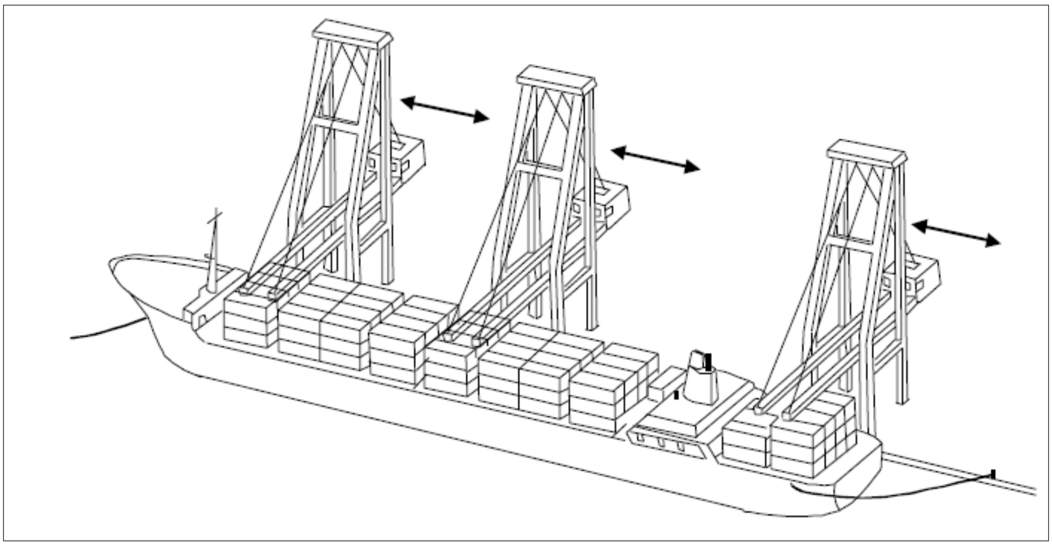


Figure 2. A drawing of QCs working on a vessel (Kim and Park (2004))

2 PROBLEM DEFINITION

The quay crane scheduling problem (QCSP) is to find a schedule for the loading and unloading tasks of a vessel by using a set of cranes such that some objective is to be minimized or maximized. In more detail, we are given a set of tasks $T = \{1,2,\ldots,nbT\}$, which are on a set of bays $B = \{1,2,\ldots,nbB\}$, and a set of assigned quay cranes $C = \{1,2,\ldots,nbC\}$ which are identical. Each task $i \in T$ must be performed by a single QC without preemption. QCs are operated on the same track; consequently, they cannot cross each other (see Fig.3). Each task has a processing time p_i which represents the time required to complete task i by any crane. The problem is to find time-intervals in which tasks are processed by the cranes with respect to a wide variety of problem constraints. Most of the time, the objective is to minimize the completion time of the latest completed job (makespan). This problem with makespan minimization is NP-hard (Lim *et al* (2007), Lee *et al* (2008)).

The definition of task divides the general QCSP into two major classes:

a) QCSP with complete bays, in whicha vessel is divided into parts longitudinally into bays. A single task consists of all unloading and loading tasks of a bay.
b) QCSP with container groups, in whicha task represents a group of containers that are stored in a bay and usually have a common destination.

To compare two major classes in the simplest way, in QCSP with completebays, the maximum number of tasks equals the number of bays. However, in QCSP with container groups, there can be more than onecontainer group in a single bay hence the number of tasks is not restricted and can be more than the number of bays. In the literature, these two classes are treated separately according to the survey by Meisel and Bierwirth (2010). In that survey, the authors investigate theberth allocation problem and QCSP literature comprehensively and develop classification schemes for both problems.

The QCSP with completebays is introduced by Daganzo (1989) andis also the first QCSP article in the literature. Peterkofsky and Daganzo (1990) solvethe mixed-integer programming model provided by Daganzo (1989) with branch and bound method.In another study that deals with complete bays, Lim *et al* (2007) show that the optimal solution can be found by searching all unidirectional schedules. In unidirectional schedules,all QCs have the same moving direction and they can only move in one direction. The authors also introduce that all unidirectional schedules can be obtained from QC-to-bay assignments. Based on this premise, they develop a constraint propagation method, simple approximation heuristics and a simulated annealing metaheuristic. However, the only constraint that they take into account is non-crossing of QCs whichmakes the problem easier to solve. Lee *et al* (2008) develop an efficient genetic algorithm to find near optimal solutions for the QCSP as introduced by Daganzo (1989).Lee and Chen (2010) identify some important deficiencies exists in the literature and develop a couple of approximation algorithms after resolving these deficiencies. However, their problem class is less detailed than the QCSP with container groups.

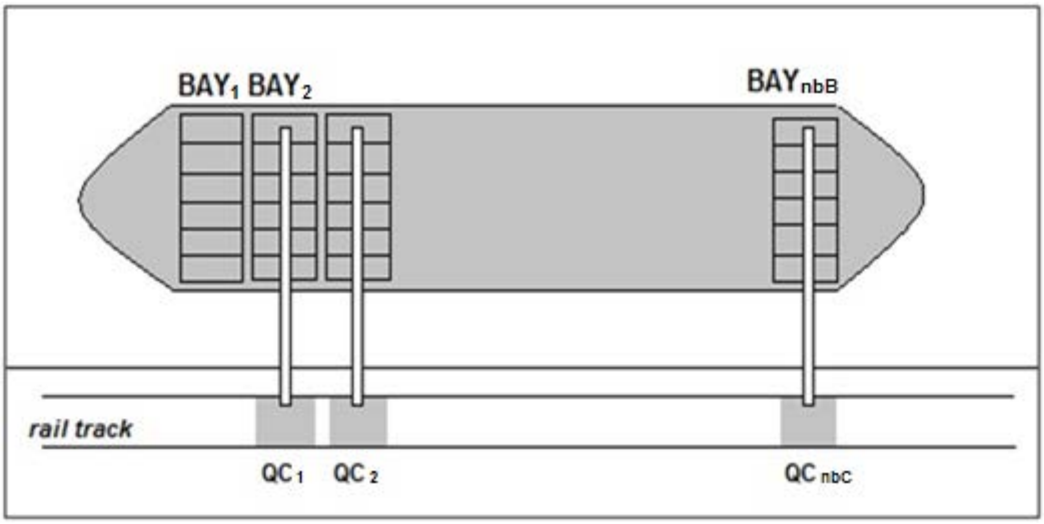


Figure 3. Positioning of QCs and bays

The QCSP with container groups, which is the most complex QCSP class among other classes,is formulated by Kim and Park (2004).The authors develop a detailed mathematical model that covers a wide variety of problem constraints.They propose a branch and bound method to solve small instances and greedy randomized adaptive search procedures (GRASP) for larger instances.The mathematical model of Kim and Park (2004) is improved by the stronger formulation of Moccia *et al* (2006). The authors develop a branch and cut method to solve the problem. Sammarra*et al* (2007) extend this study byidentifying some interference among QCs and solve the modified QCSP model by using the tabu search metaheuristic.Bierwirth and Meisel(2009) alsoinvestigate QC interference constraints and then develop a branch and bound based heuristic solution procedure for the problem. The authors also show that the optimal schedules of QCSP with container groups do not have to be unidirectional schedules.Meisel and Bierwirth (2011) introduce a unified approach to compare different problem classes and characteristics. Moreover they provide a scheme for generating benchmark instances with certain characteristics.Legato *et al* (2012) solve the problem by considering independent unidirectional schedules, where cranes can have individual moving directions.

In this study, the class of QCSP with container groups is studied based on the research stream started by Kim and Park (2004) and a constraint programming model is developed by considering deficiencies previously identified by Bierwirth and Meisel(2009), and Lee and Chen (2010).

3 METHODOLOGY

In this section, some concepts related to the methodologies used in this study are introduced.

3.1 Constraint Programming

Constraint programming (CP) is a technique that can be used for representing and solving combinatorial problems which are hard to solve. Similar to mathematical programming, CP is a combination of defining constraints about the problem via variables and finding a solution that satisfies all the constraints. However, in CP, constraints are used actively to infer new constraints and to reduce domains of variables by removing values.

Constraint programming also provides rich modeling tools to represent complex combinatorial problems in a very compact way. Every constraint has its propagation algorithm so that while creating a CP model, at the same time, which propagation algorithmsto be usedin solving the problem is also decided.Differently from mathematical programming,the main focus of CP is on constraints and the feasibility, rather than the objective function and the optimality.

3.2 Intervaland Sequence Variables

The QCSP mustbe represented in terms of activities and resources to define some concepts which are used in the proposed CP model.In this representation, the tasks to be performed by quay cranes correspond to activities and quay cranes correspond to resources. Then each activity A_i has a start time $start(A_i)$,an end time $end(A_i)$ and a processing time $p(A_i) = end(A_i) - start(A_i)$.

An interval variable (Laborie and Rogerie(2008)) is an interval of time during which an activity is executed. The decision here is to select when to execute this activity during the planning horizon. Each interval variable is characterized not only by a start time, an end time and a processing time but also with presence information, $presence(A_i)$.An interval variable can be optional;therefore, the activity can be left unexecuted. In these situations, unexecuted activitytakes$presence(A_i) = False$.On the other hand, all executed optional tasks take $presence(A_i) = True$.This property of an interval variable helps to model when there are activities that can be executed on a set of alternative resources. In this study, a task can be executed on a set of QCs; therefore, optional interval variables are used.

A sequence variable (Laborie *et al* (2009)) is a decision variable that its value is a permutation of some group of interval variables.Sequence variables can also keep transition times between interval variables.In this study, this property is used to implement travel times of QCs. We can simply construct the same model without using any sequence variable; however,it brings more effective constraint propagation. Hence,sequence variables areused in this study to implement travel times.

3.4 Global Constraints

A global constraint inCP can represent the complicated relationships between problem variables as a single constraint.Usually it captures most of the problem variables, therefore provides faster and effective domain reduction by usingspecialized pruning algorithms. The most widely known global constraint is$alldifferent(v_1,\ldots,v_n)$,where $v_i, i \in \{1, \ldots, n\}$, is the decision variable with $domain(v_i) = [d_1, \ldots, d_n]$. It ensures that each variable$v_i$must take a different value from its domain.On the other hand, $n(n-1)/2$ constraints are required to represent such a relationship in a mixed-integer programming model.

There are many different global constraints in the literature and two of them are considered in this study.The $alternative(\forall A_i, Y_{ij})$ constraint (Beck and Fox (1999)) simply assigns eachactivity to a single resource. The property of optionality of the interval variables makes sense here. Assume that there are m resources in the problem and interval variable $Y_{ij'}$ indicates that activity i is assigned to a resource j'. Since each activity is assigned to only one resource, Y_{ij} variables are not calculated while $j \neq j'$.Furthermore a $disjunctive(\forall A_i | A_i \in P)$ global constraint ensures that the activities which areelements of some set P, should not overlap.

4 MODELING

As previously noted in the problem definition section, in this study QCSP with container groups is studied and from now on, QCSP will refer to QCSP with container groups. In this section, a constraint programming model for QCSP with container groups is proposed. After representing a new lower bound generation scheme in Section 4.5, lastly in Section 4.6,the benefits of constraint programming for QCSP arediscussed.

4.1 Assumptions

The following assumptions are made for the QCSP:

1. Every task must be completed by a single crane and tasks are non-preemptive.
2. Quay cranes are identicaland are operated on the same track.
3. In a single bay, only one QC can work because of its size.
4. In any schedule,presence of positions of idle QCs must be considered.Bierwirth and Meisel (2009) and Lee and Chen (2010) both identified that idle cranesshould not be ignored.
5. Two QCs cannot be operated simultaneously in adjacent bays; therefore there must be one bay of safety margin between two adjacent QCs at any time.
6. Travel time of a QC between two adjacent bays is one time unit.
7. Allocated rail-track width must be equal to vessel size. This means that a crane assigned to a vessel cannot travel out of the boundary of this vessel.
8. Initial positions of QCs are ignored in this study. Initial positions of QCs may be essential to generate unidirectional schedules, but in any non-unidirectional method, some good solutions may be restricted by using pre-definite initial positions. Since an optimal solution for an instance is directly dependent on initial positions of QCs, in this study, a starting position of a QC is set to be the bay that it starts processing tasks in the schedule. That is, we are generating a schedule without defining initial positions, and then QCs are starting from the bay of the first scheduled task.

4.2 Sets and Parameters

nbT number of tasks of the vessel to be processed;

nbB number of bays in the vessel;

nbC number of quay cranes assigned to the vessel;

T set of tasks, $T = \{1, \dots, nbT\}$;

B set of bays, $B = \{1, \dots, nbB\}$;

C set of quay cranes, $C = \{1, \dots, nbC\}$;

p_i processing time of task $i, \forall i \in T$. Each quay crane is identical, therefore processing time for task i is same for all quay cranes;

$Prec$ set of precedence among tasks, i.e. $Prec = \{< i, k >, i, k \in T\}$;

l_i bay of task $i, \forall i \in T$;

TP sum of processing times of all tasks, i.e. $\sum_{i \in T} p_i$;

S_b set of tasks at bay b, i.e.$S_b = \{\forall i \in T: l_i = b\} \forall b \in B$;

$early_i$earliest starting time for task i; that is,total workload of predecessorsof task i, i.e. $early_i = \sum_{<k,i> \in Prec} p_k \; \forall i \in T$.

4.3 Decision Variables

X_i an interval variable that represents the interval in which task i is processed (by any quaycrane),$domain(X_i) = [early_i, TP], \forall i \in T$;

Y_{ij} an optional interval variable that represents the interval in which task i is processed by quay crane j,$domain(Y_{ij}) = [early_i, TP], \forall i \in T, \forall j \in C$;

Z an integer variable that defines the makespan, i.e. the maximum completion time of all tasks by quay cranes, $domain(Z) = [LB, TP]$. (the procedure to generate lower bound LB is represented in the next section)

A proper model can be constructed by using only these decision variables above; however, a set of sequence decision variables is added to the model to strengthen the inference among problem elements,as follows:

$SeqC_j$ a sequencevariable that keeps the permutation of the tasks to be processed by QC j, $domain(SeqC_j) = \{permutation\ of\ Yij |\ i \in T\ and\ presence(Y_{ij}) = 1\}, \forall j \in C$.

4.4 CP model

The constraint programming model is formulated as follows:

$minimize\ Z$

Subject to:

$Z = max_{\forall i \in T, \forall j \in C}\ (end(Y_{ij}))$ (1)

$alternative\left(X_i, \left(Y_{ij} \middle| \forall j \in C\right)\right)$ $\forall i \in T$ (2)

$disjunctive(Y_{ij} | \forall i \in T)$ $\forall j \in C$ (3)

$disjunctive(Y_{ij} | \forall i \in S_b, \forall j \in C)$ $\forall b \in B$ (4)

$$disjunctive\left(Y_{ij}, Y_{nm}\middle|\ \forall i \in S_b, \forall n \in S_{b+1}\ \forall j, m \in C\ \right) \qquad \forall b \in\ B - \{nbB\} \qquad (5)$$

$$(l_i > NbB - NbC + i) \vee (l_i < j) \rightarrow\ presence\left(Y_{ij}\right) = 0 \qquad \forall i \in T, \forall j \in C \qquad (6)$$

$$disjunctive(Y_{ij}, Y_{nm}\ ) \qquad \forall i, n \in T, \forall j, m \in C, l_i \geq j, (j \neq nbC), (l_i \neq nbB), (l_n \geq m), (l_n \geq l_i), (m \geq j), (l_n < l_i + 2(m - j - 1) + 2) \quad (7)$$

$$end\left(Y_{ij}\right) \leq start(Y_{kr}) \qquad \forall < i, k > \in Prec, \forall j, r \in C, (i \neq k) \qquad (8)$$

$$presence\left(Y_{ij}\right) \wedge presence(Y_{nm})\ \ \rightarrow$$
$$\left(\ end(Y_{nm}) + l_n - l_i - 2(m - j) \leq start\left(Y_{ij}\right)\right) \veebar \left((end\left(Y_{ij}\right) + l_n - l_i - 2(m - j) \leq start(Y_{nm})\right)$$
$$\forall i, n \in T, \forall j, m \in C, (l_i \geq j), (m > j), (l_n < l_i + 2(m - j)), (l_n \geq m) \qquad (9)$$

$$presence\left(Y_{ij}\right) \wedge presence(Y_{nm})\ \ \rightarrow$$
$$\left(\ end(Y_{nm}) + l_n - l_i + 2(j - m) \leq start\left(Y_{ij}\right)\right) \veebar \left((end\left(Y_{ij}\right) + l_n - l_i + 2(j - m) \leq start(Y_{nm})\right)$$
$$\forall i, n \in T, \forall j, m \in C, (l_i \geq j), (m < j), (l_n > l_i - 2(j - m)), (l_n \geq m) \quad (10)$$

$$presence\left(Y_{ij}\right) \wedge presence\left(Y_{nj}\right)\ \rightarrow$$
$$\left(\ end\left(Y_{nj}\right) + |l_i - l_n| \leq start\left(Y_{ij}\right)\right) \veebar \left((end(Y_{ij}) + |l_i - l_n| \leq start\left(Y_{nj}\right)\right)$$

$$\forall i, n \in T, \forall j, m \in C, (l_i \geq j), (j \neq nbC), (l_i \neq nbB), (l_n \geq m), (l_n \geq l_i), (m \geq j), (l_n < l_i + m - j) \quad (11)$$

$$disjunctive(SeqC_j) \qquad \forall j \in C \quad (12)$$

The objective function is to minimize the completion time of the latest QCwhich is calculated by constraint (1). Constraint set (2) is a global constraint to assign each task to one and only one QC. The following two constraints are global constraints to forbid definite tasks to overlap. Global constraint set (3) ensures that tasks which are assigned to the same QC will not overlap. Also global constraint set (4) avoids the interferenceamongQCs by not allowing the overlap of the tasks which are located in the same bay. Constraint set (5) ensures that two tasks that are located in adjacent bays cannot be processed simultaneously; that is, the safety margin is assumed to be one bay. Constraint sets (6) and (7) resolve two deficiencies that are identified by Lee and Chen(2010). Constraint set (6) is defined to keep QCs within the boundaries of the vessel. Constraint set (7) ensures that there will always be enough space between two cranes to accommodate in-between cranes. Precedence relations among tasks are defined by constraint set (8). Constraint sets (9) and (10) together ensure that QCs cannot cross each other since they are operated on the same track. Travel times of QCs are implemented in constraint set (11). Constraint set (12) ensures that members of a defined sequence will not overlap by also taking travel times into account. Constraint set (11)is sufficient to represent travel times individually; however, (11) and (12) are used together for better propagation. The correct treatment of travel times and safety margins identified by Bierwirth and Meisel (2009) are also embedded in (9) and (10).

4.5 Lower Bound for QCSP

Generating tight lower bound values for QCSP instances is helpful because of two main reasons. First of all, a tight lower bound allows determining solution quality of the CP results more accurately. Moreover, a tight lower bound helps the CP model to terminate earlier with an optimal result, when $CP\ result = lower\ bound$.

In this study, the results show that most of the time the simplest parallel-machine scheduling lower bound of TP/nbC provides considerably near values to the optimal. However, in some instances the gap with this simple lower bound reaches 20%. It is observed that an optimal result can be significantly more than that because of excessive workload of some bays and/or some operational restrictions caused by safety distances. After this observation, a simple yet powerful mixed-integer programming model was developed to find tight lower bounds (LB) for QCSP with container groups.In this relaxed QCSP model, all sets and parameters are the same with those of the proposed CP model. Additional parameters and decision variables are represented below:

R_{ij} a binary decision variable that takes value 1 if task i is assigned to $QC\ j$, otherwise 0, $\forall i \in T, \forall j \in C$;

A_{jb} a binary decision variable that takes value 1 if $QC\ j$ is assigned to at least one task of bay b, otherwise 0, $\forall b \in B, \forall j \in C$;

LB an integer decision variable that represents the makespan;

$Load_b$ a parameter that calculates the total workload of two adjacent bays; that is, $Load_b = \sum_{i \in S_b} p_i + \sum_{i \in S_{b+1}} p_i \quad \forall b \in B - \{nbB\}$;

M a very large number.

Then, mixed-integer programming model is formulated as follows:

minimizeLB

Subject to:

$$\sum_{j \in C} R_{ij} = 1 \qquad \forall i \in T \qquad (13)$$

$$LB \geq TP/nbC \qquad (14)$$

$$R_{ij} = 0 \qquad \forall i \in T, \forall j \in C, l_i > NbB - 2(NbC - j)\ \ or\ l_i < 2j - 1 \qquad (15)$$

$$\sum_{i \in T} R_{ij} < MA_{jb} \qquad \forall j \in C, \forall b \in B \qquad (16)$$

$$LB \geq \sum_{i \in T, j \in C} R_{ij} p_i + \sum_{j \in C, b \in B} A_{jb} - 1 \qquad (17)$$

$$LB \geq max(Load_b, \forall b \in B - \{nbB\}) + 1 \qquad (18)$$

$$R_{ij} \geq 0, A_{jb} \geq 0 \qquad \forall i \in T, \forall j \in C, \forall b \in B \qquad (19)$$

In this model, each task is assigned to a QC by constraint set (13) while minimizing the workload of the densest QC. Constraint (14) defines the parallel-machinescheduling pre-emptive lower bound.Task-to-QC assignments are also restricted by the constraint set (15) that allocates the rail-track area. In order to addtravel times in the simplest form, QC-to-bay assignments are tracked for each QC by using two-indexed binary decision variables in (16). Constraint (17) defines one lower bound with respect to the workload of QCs. If a QC is assigned to m different bays, then $(m - 1)$ time unit is added to the total workload of this QC.

$(m-1)$time units are added rather than m because the initial position of QC and one of the assigned bays can be the same; therefore, initial travel of this QCbefore starting to process assigned tasks will not be needed. Also safety margin requirements are considered from a different, but a simpler perspective. It is not possible to process two bays simultaneously with two different QCs because of the safety margin requirements. If we first list total workloads of each bay and then select the maximum total workload of two adjacent bays, it will give us another potential lower bound for the instance. Call this value V, then LB is always greater than or equal to V+1 because of the safety margin requirements; that is, if these two adjacent bays are processed by the same QC, one unit of travel time is required for that QC to travel between these bays. On the other hand, if these two are processed by two different QCs, the second QC cannot start processing the second bay before the first QC leaves from the other bay because of the safety margin. Therefore, again one time unit is required. Constraint (18) defines this lower bound.

Non-crossing, non-interference and precedence constraints are not considered in this MIP model, hence it represents a relaxed version of the QCSP. Nevertheless, the model generates and tight lower bounds for QCSP quickly. For example, in one of the instances of Meisel and Bierwirth (2011)$TP/nbC$is 1000 and the proposed LB is 1174. Note that the best observed result for this instance is also listed as 1174 in their study. In addition, the MIP model is solved optimally in half a minute, at worst. For most of the instances, it is solved in less than5 seconds.

4.6 Advantages of using CP for QCSP

In constraint programming context,$nbT\ (nbC+1)+1$variables are enough to represent the whole problem, while correspondingmixed-integer programming model has $nbT^2(nbC+1)+3nbTnbC+3nbC+nbT+1$ variables. We can see the effects of these numbers with an instance of 20 bays, 30 tasks and four quay cranes. For this instance,the proposed CP model and the MIP model have 151 and 4903 decision variables, respectively.This significant difference is mainly due to synthesis of rich modeling tools of CP and efficient modeling. On the other hand,there is no significant difference between the models with respect to the number of constraints. However, most of the constraints only consist of Y_{ij} variables in the CP model.The fact of having a very small number of variables and large number of constraints with strong relationshelps the constraint programming model work effectively for QCSP.

In real-world problems, it is possible that additional constraints exists other than presented here. For example, quay cranes may have certain time-windows due to different reasons according to Meisel (2011). Another example is the case that task i has to be completed before a given time because there is another vessel in the terminal which is urgently waiting to receive task $i$before departure.The wide variety of such additional constraints can be easily added to the proposed CP model. Most of the time, however, such additional constraints will not make the problem harder to solve by using CP. Each new constraint that consists of existing variables may probably help to strengthen the inferences and prunes more domains, and consequently, will reduce the size of the search space. Therefore, solving more complex problems by CP will probably be easier.Hence, constraint programming can be anappropriate method to solve QCSP.

5 DESIGN OF COMPUTATIONAL EXPERIMENTS

Computational experiments were conducted to test the performance of the proposed CP model for QCSP with container groups. In this study, QCSPgen (Meisel and Bierwirth(2011)) is used to generate test instances. QCSPgen is a benchmark instance generator for QCSP that allows comparing different models and solution procedures.This generatoralso provides the most realistic benchmark instances in the literature.Detailed information about QCSPgen and the benchmark sets created by this software can be found in Meisel and Bierwirth (2011).

In this study, instances in Set B of Meisel and Bierwirth (2011) areused to test the performance of the proposed CP model since set B consist of proper instances for the real-world cases. In all Set B instances, the number of bays (nbB) and number of QCs (nbC) were taken as 15 and 4, respectively. Also six different numbers of tasks (nbT) were selected as 45, 50, 55, 60, 65 and 70. For each number of tasks,ten different instances were generated by using random seeds 1 to 10, in line with the literature. Therefore a total of 60 QCSP benchmark instances were generated by QCSPgen. Since the search phase of CP has randomness (Laborie and Godard (2007)), 10 trials with different random seeds were run for each instance.

QCSP is one of the problems to be solved frequently in container terminals. Therefore it is important to find a good solution in a short time. Even though CP is a powerful approach for QCSP with respect to the problem structure, it may have a disadvantage as CP may not result in an optimal solution within a reasonable time. Hence, CP is not used as an exact method in our study, which brings the question of when to terminate a CP search. We let CP search the solution space to find a feasible solution F with its objective function value F' and, each time a better solution is found, the constraint $Z \leq F'$ is added to the model. Obviously, when F' is equal to the lower bound (LB), the search is terminated with an optimal solution. Otherwise, CP search will be stopped after ($1.5nbT/10$) minutes with the best solution found so far.

The results of theinitial computational experiments were compared with the results for Set B instances, provided by Meisel and Bierwirth(2011) and Legato *et al*(2012).

6 RESULTS

IBM ILOG's CP Optimizer 12.3 and CPLEX 12.3 were used to solve constraint programming and mixed-integer programming models, respectively. All tests wereconducted on a computer with 2.53 GHz processor.

First, to show that the proposed CP model is a viable tool for solving QCSP, the performance of CP was demonstrated by considering the percentage derivation of CP results from the lower bound LB.

$$GAP_{CP-LB}\% \;=\; 100\times(\overline{CP\ result} - \overline{LB})\ /\ \overline{LB}.$$

Since CP was run for 10 trials with different random seeds for each instance, the $CP\ result$means the average result of these 10 trials. Moreover,there are six different instance sets in this study with different sizes; each of them consists of ten instances with same size. Hence, $\overline{CP\ result}$ and $\overline{LB}$ indicate the average values of 10 instances of the set.

Second, we compared $\overline{LB}$ with $\overline{LB_{M\&B}}$, which indicates the average lower bound values produced by Meisel and Bierwirth (2011). $\overline{LB_{M\&B}}$values were derived by CPLEX within 2-hour time limit.

$$GAP_{LB}\% \;=\; 100\times(\overline{LB} - \overline{LB_{M\&B}})/\ \overline{LB_{M\&B}}\ .$$

After showing CP model's success on finding near optimal solutions, we compared the performance of the proposed CP model toUDS heuristic (Bierwirth and Meisel (2009)) and

TPN procedure (Legato *et al* (2012)). Hence, other major performance is defined as the percentage gap between average solution times of CP model ($timeCP$) and TPN procedure ($timeTPN$). We disregarded the gap between solution times of our method and UDS heuristic because the TPN procedure surpasses UDS heuristic at each instance size:

$$GAP_{time}\% = 100\times(\overline{timeCP} - \overline{timeTPN}) / \overline{timeTPN}.$$

Lastly, we also present the average standard deviation of CP results. Results of the computational experiments are presented in Tab. 1 below.

Table 1. Results of computational experiments

$Instances$		$Lower\ Bounds$		%	Z				$Time\ (minutes)$			%	%
nbT	nbC	$\overline{LB_{M\&B}}$	$\overline{LB}$	GAP_{LB}	UDS	TPN	CP	σ_{CP}	UDS	TPN	CP	GAP_{time}	GAP_{CP}
45	4	754.3	770.5	2.15	775.8	775.8	773.4	0.72	11.88	5.73	5.27	-8,03	0.38
50	4	753.4	763.1	1.29	770.9	770.9	769.3	0.70	20.85	13.78	6.8	-50.65	0.81
55	4	753.6	767.1	1.79	771.9	771.9	771.8	0.94	17.97	10.36	7,38	-28.76	0.61
60	4	753.1	764	1.45	771.1	771.1	770.9	1.36	21.95	22.47	8,07	-64.09	**0.90**
65	4	753.5	765.9	1.65	769.0	769.0	768.7	1.01	35.3	19.6	8.22	-58.06	0.37
70	4	753.1	756.3	0.42	762.1	761.9	761.3	1.03	37.18	22.59	8.9	-60,6	0.66
$Averages$		753.5	764.48	1.46	770.1	770.1	769.2	0.96	24.19	15.76	7.44	-52,78	0.62

New lower bound LB which is generated in at worst 30 secondsprovides 1.46%tighter bounds for the problem. Therefore, the average GAP_{CP} value of 0.62% indicates that the proposed CP model is convenient to solve QCSP. The maximum average gap is observed for $nbT = 60$ and it is 0.9%. Furthermore, the low standard deviation (σ_{CP}) values over 10 trials indicate the robustness of the CP approach. During computational experiments, 93 times in 600 total trials,the search was terminated by the optimal result within time-limit, mainlyby the help of new tighter lower bound values.

Moreover, the overall solution quality of the proposed CP model is almost similar to the UDS and TPN methods. An insignificant advantage of our results is probably caused by the inexistence of pre-defined initial positions in this study

The proposed CP model also cuts the previous best solution time by 52.78%. In other words, our CP methodsolves the problem approximately two times faster than previous best,which is a significant improvement for the solution of QCSP.In just a very few instances of Set B, non-unidirectional schedules provides considerable (more than 1%) improvement to unidirectional results.Therefore, we can state that overall solution quality of unidirectional schedules (or independent-unidirectional schedules) is almost the same with the non-unidirectional schedules.

7 CONCLUSION

In this study, a constraint programming model is proposed for QCSP with container groups. By using rich modeling tools of the CP, the number of variables in the QCSP is reduced by an order of number of tasks. For applicability of any QCSP study to real-world problems, constraints of these problems should be reflected properly into the model. Therefore some deficiencies, which were identified in Lee and Chen (2010), are fixed for the QCSP with container groups class. Also the travel times of QCs and safety margins between adjacent QCs are implemented with the corrections of Bierwirth and Meisel(2009).

The computational experiments show that the proposed CP model is a fast andaccurate alternativeto obtain near optimal solutions for QCSP. The solution quality of the proposed CP model is parallel to the most efficient solution methods in the literature; however, at the same time, our method has a significant solution time advantage over these methods. Moreover, we are going to test the model with more instances with different characteristics to be able to draw better conclusions. Our aim is to develop a model that not only works fast and is flexible for all sizes and types of practical instances, but also applicable in real-world container terminals.

REFERENCES

Beck, J. C., and Fox, M. S. (1999), Scheduling alternative activities,*Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, Florida, USA, 18-22, July 1999.

Bierwirth, C. and Meisel, F. (2009),A Fast Heuristic for Quay Crane Scheduling with Interference Constraints,*Journal of Scheduling*, **12**, 345-360.

Bierwirth, C. and Meisel, F.(2010),A Survey of Berth Allocation and Quay Crane Scheduling Problems in Container Terminals,*European Journal of Operational Research*,**202(3)**, 615-627.

Daganzo, C.F.(1989), The Crane Scheduling Problem, *Transportation Research B*, **23(3)**,159-175.

Guinet, A. (1993), Scheduling Sequence-Dependent Jobs on Identical Parallel Machines to Minimize Completion Time Criteria,*International Journal of Production Research*, **31(7)**, 1579–1594.

Kim, K.H. and Park, Y.M. (2004),A Crane Scheduling Method for Port Container Terminals, *European Journal of Operation Research*,**156**,752–768.

Laborie, P. and Godard, D. (2009), Self-Adapting Large Neighborhood Search: Application to single-mode scheduling problems, *Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications*, Paris, France,28-31, August 2007.

Laborie, P. and Rogerie, J. (2008), Reasoning with Conditional Time-intervals,*Proceedings of the 21st International FLAIRS Conference*,Coconut Grove, Florida, USA, 15-17, May 2008.

Laborie, P., Rogerie, J., Shaw, P. and Vilim, P. (2009),Reasoning with Conditional Time-intervals - Part II: an Algebraical Model for Resources,*Proceedings of the 22nd International FLAIRS Conference*, Sanibel Island, Florida, USA, 19-21,May 2009.

Lee, D-H and Chen, J.H. (2010), An Improved Approach for Quay Crane Scheduling with Non-Crossing Constraints, *Engineering Optimization*, **42(1)**, 1-15.

Lee, D-H, Wang, H.Q. and Miao,L. (2008), Quay Crane Scheduling with Non-Interference Constraints in Port Container Terminals,*Transportation Research E*, **44**, 124–135.

Legato, P., Trunfio, R. and Meisel, F. (2012), Modeling and Solving Rich Quay Crane Scheduling Problems,*Computers & Operations Research*, **39**, 2063–2078.

Lim, A., Rodrigues, B. and Xu, Z.(2007), A m-Parallel Crane Scheduling Problem with a Non-Crossing Constraint,*Naval Research Logistics*,**54(2)**, 115–127.

Meisel, F. (2011), The Quay Crane Scheduling with Time Windows, *Naval Research Logistics*, **58(7)**, 619-636.

Meisel, F. and Bierwirth, C. (2011),A Unified Approach for the Evaluation of Quay Crane Scheduling Models and Algorithms, *Computers & Operations Research*, **38**, 683 –693.

Moccia, L., Cordeau, J.F., Gaudioso, M. and Laporte, G. (2006), A Branch-and-Cut Algorithm for the Quay Crane Scheduling Problem in a Container Terminal, *Naval Research Logistics*, **53(1)**, 45–59.

Peterkofsky, R.I. and Daganzo, C.F.(1990), A Branch-and-Bound Solution Method for the

Crane Scheduling Problem,*Transportation Research Part B*,**24(3)**, 159–172.
Sammarra, M., Cordeau, J.F., Laporte, G. and Monaco, M.F.(2007), A Tabu Search Heuristic for the Quay Crane Scheduling Problem, *Journal of Scheduling*, **10(4-5)**, 327–336.
Stahlbock, R. and Voß, S. (2008), Operations Research at Container Terminals: a Literature Update, *OR Spectrum*, **30(1)**, 1–52.
Steenken, D., Voß, S. and Stahlbock, R. (2004), Container Terminal Operation and Operations Research – a Classification and Literature Review,*OR Spectrum*, **26(1)**,3–49.
Vis, I.F.A. and de Koster, R. (2003), Transshipment of Containers at a Container Terminal: an Overview,*European Journal of Operational Research*, **147(1)**, 1–16.

ECONOMIC SHIP TRAVEL SPEED: CONSEQUENCES FOR MARITIME LOGISTIC STRATEGIES AND SUSTAINABILITY

Herbert Kotzab[1] and Timm Gudehus[2]

[1]*Dept. of Logistic Management, University of Bremen, Wilhelm-Herbst-Strasse 12, 28359 Bremen, kotzab@uni-bremen.de; CBS International Professor Glasgow Caledonian University*

[2]*Strandweg 54, 22587 Hamburg, TimmGudehus@t-online.de*

ABSTRACT

In this paper we show how an economic travel speed of cargo ships impacts the profit situation as well as the environmental sustainability significantly. We thereby differ between a cost-optimal and profit-optimal ship speed strategy and show, based on model calculations, how both strategies lead to lower costs as well as lower emissions. Following the dynamic network aspect, we suggest that large container ship companies can adapt both strategies under specific market conditions and allow them to act profitable as well as environmental sustainable.

Keywords: Maritime logistics, cost-optimal speed, profit-optimal speed, sustainability

1 INTRODUCTION

The operative task of maritime logistics is to convey cargo with ships on rivers, channels and seas at minimal possible costs, fuel consumption and emissions. For this purpose optimal shipping networks and maritime transport chains have to be designed, implemented and operated (see Gudehus and Kotzab, 2012).

Optimality, besides ship capacity, is however driven in by additional economic as well as environmental factors such as (see e.g. Notteboom and Vernimmen, 2008; Psafaratis and Kontovas, 2010)

- extreme market volatility in regards to oil prices and freight rates thus leading to extreme demand up- and downturns;
- high dependency of travel speed on fuel consumption leading to a high contribution of bunker costs to the total operating costs;
- huge greenhouse gas emissions (GHG) emissions.

Especially travel speed is impacting on the economic as well as the sustainable optimality of maritime logistics (see Fig. 1).

The example of slow steaming shows that fuel consumption can be significantly decreased as well as emissions of GHG (see e.g. Cariou, 2011; Eyring et al., 2010; Faber et al. 2010; Song and Xu, 2012). For this paper we determine a cost-optimal speed and profit-optimal speed of ships that significantly differ in their economic as well as environmental results.

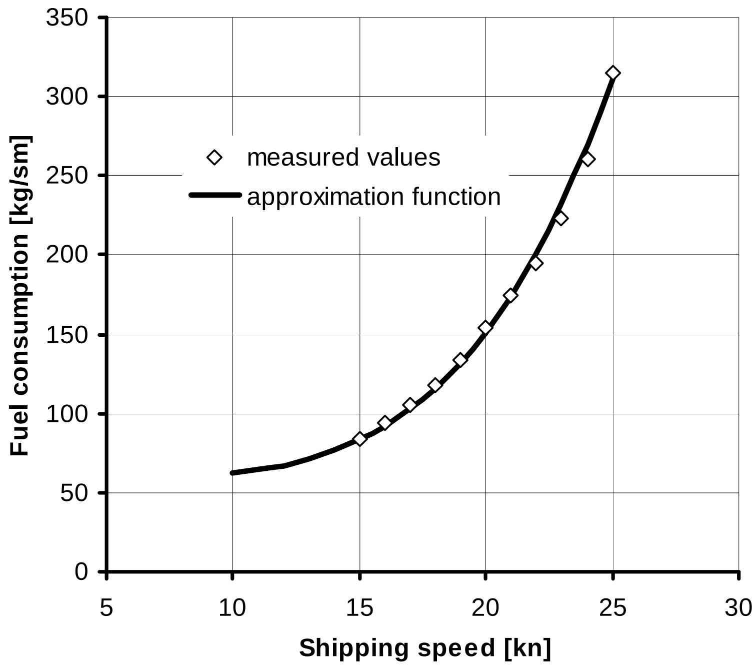


Figure 1. Fuel consumption curve of a 5000-TEU-container ship**.** *Approximation function*: $c_F(v) = 58 + 0{,}00013 \cdot v^{4,5}$, (Gudehus and Kotzab 2012, 824)

2 INFLUENCING PARAMETERS ON MARITIME LOGISTIC COSTS

2.1 Fuel Consumption and Bunker Costs

The crucial element for determining the optimal speed is the dependency of the mileage consumption on the travel speed of a ship (e.g. Gudehus 1964 and 1967;; Ronen, 1982; Notteboom and Vernimmen, 2008; Eyring et al., 2010; Gudehus 2010). As shown in Fig. 1 for a 5,000 TEU-containership, the fuel consumption curve increases from a basic consumption at minimal operating speed up to the highest consumption at maximal speed for which a ship has been designed. The slope of the bunker consumption curve then depends on the type, hull form, propulsion, capacity and other properties of the ship as well as on load, weather and sea conditions (see also Schneekuth and Bertram, 1998; Faber et al., 2010).

Generally, it holds that the mileage consumption of a cargo ship increases with a certain power n of the travel speed, which is in the range between 4 to 6 (Gudehus 2010). In case a ship travels a tour, the consumption for fuel increases due to different speeds during the tour as well as to accelerations. In order to save fuel it is advisable to let a ship travel as constantly as possible in all sections with an average speed.

Looking at the bunker costs, it is known that bunker quantities are purchased at different bunker prices during a tour. In case of constant bunker prices for the whole tour, the total bunker costs are proportional to the total fuel consumption and minimal when travelling with constant speed.

2.2 Transport Time and Freight Limit Performance

The transport time between two harbours is the sum of the harbour time spent at each called harbour and the travel time between these harbours.

The harbour time is the sum of all times for decelerating, pulling in, landing, loading and un-loading, pulling out, and acceleration, and of the waiting times (see e.g Notteboom and Vernimmen, 2008).

The times required for locking, channel passing and other interruptions can be taken into account similar to the harbour times. When scheduling a tour, the actual stop times at the different harbours and other interruptions must be known. However, these times do not alter the cost-optimal speed and have only minor influence on the profit-optimal speed.

Transport time and travel speed are interrelated. A reduction in travel speed prolongs transport time nearly in the same relation. Any significant increases of the transport time may thereby affect the achievable freight rates. Especially when high value cargo is shipped, we need to take – at least – additional interest costs into account.

In order to utilize a given ship capacity best, we need to consider a maximally achievable filling degree. This can be obtained by optimal stowage plans. However, due to the many stowing restrictions of container ships in practice the achievable filling degree can be much lower than 95 %, whereas for bulk carriers also higher filling degrees than 95 % are possible (see Gudehus and Kotzab, 2012).

Model calculations have shown that a 20 % reduction of speed can reduce the freight limit performance of a ship, which is the theoretical maximal freight capacity, by 18 %. As long as the sum of the travel times is significantly greater than the sum of the harbour times, the freight limit performance of the ship increases and decreases in proportion with the travel speed.

2.3 Shipping Operating costs and Shipping Freight Costs

The operating costs for a ship refer to the costs for bunker, the costs for utilizing the ship and the harbour costs (see e.g. HSH Nordbank, 2009). Bunker costs depend on the bunker costs per roundtrip, the cycle frequency and the operation time. The utilization costs are determined by the operation time and ship utilization price, which is for own ships the cost rate and for leased ships the charter rate.

The ship utilization price increases with the capacity and the installed maximal speed. As far the technical utilization time of the ship and the lubricant consumption do not significantly alter with the speed, the utilization costs are independent from the current travel speed.

The harbour costs depend on operation time, number of stops per tour and the average harbour price which is the sum of the charges and dues for harbour facilities, pier utilization, tow boats, pilots, waterway utilization and other local services. Fees for channel passages, locks and other intermediate services can be treated like harbour prices. The harbour prices vary from stop to stop but do not depend on the travel speed.

The shipping freight costs per load unit are the ship operating costs divided by the current freight performance of the ship. Their dependence on the shipping speed is shown in Fig. 2. The freight costs are minimal at the cost-optimal speed, which in this example is 15.3 kn. As long as the freight demand for all segments of the tour exceeds the freight limit performance, the current freight performance equals the freight limit performance of the ship. Thereby we need to acknowledge that freight costs decrease at first with increasing speed and, after passing a flat minimum at the cost-optimal speed, sharply increase.

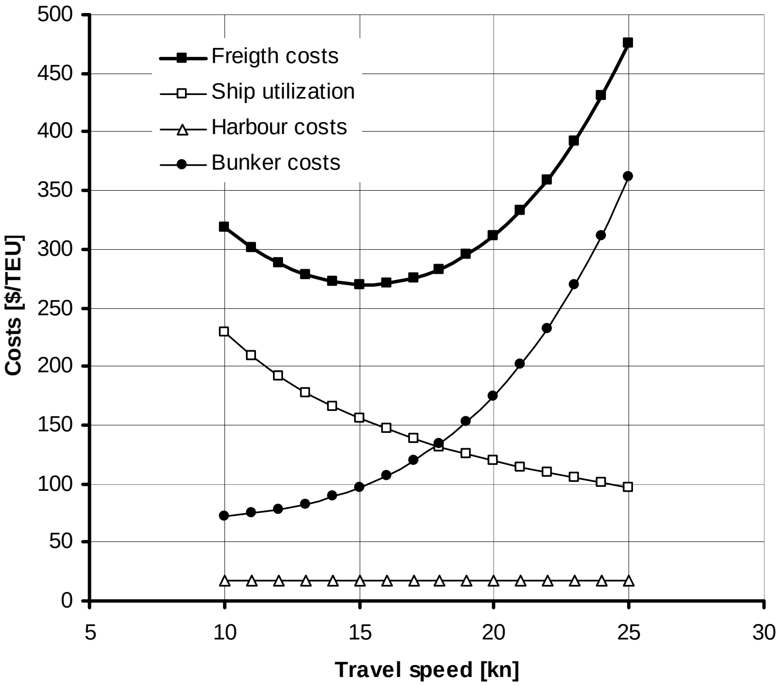


Figure 2. Contributions and speed dependence of the shipping freight cost for full utilization of the effective capacity

2.4 Operating Profits

It is of course the goal when operating a shop to achieve a return which covers the business operating expenses of the shipping company and generates maximal profit. The operating profits of a single ship are the difference between the total freight revenues generated by the freight performance in the considered operation time and the operating costs (see also Corbett et al., 2009). We can thereby observe that a decrease in travel speed leads to an increase in operating profits as compared to a travel at maximal speed.

The resulting influence of the shipping speed on the operating profit is shown in Fig. 3. In this case, the operating profit is maximal at the profit-optimal speed 20.7 kn.

3 DETERMINATION OF ECONOMIC SHIP TRAVEL SPEEDS[1]

3.1 Cost-optimal Speed

The cost-optimal speed can be determined by looking at the ratio between ship utilization price and the bunker costs times the ascent parameters of the fuel consumption curve. It can be observed that the cost-optimal speed of a cargo ship increases with the (n+1)-root of the ship utilization price and decreases inverse proportional with the (n+1)-root of the bunker price. Furthermore, we can see that the cost-optimal speed is independent from the ship capacity and the tour length as well as from the number and costs of the harbour stops.

[1] A detailed derivation and discussion of the presented model can be found in Gudehus and Kotzab (2012, pp. 823):

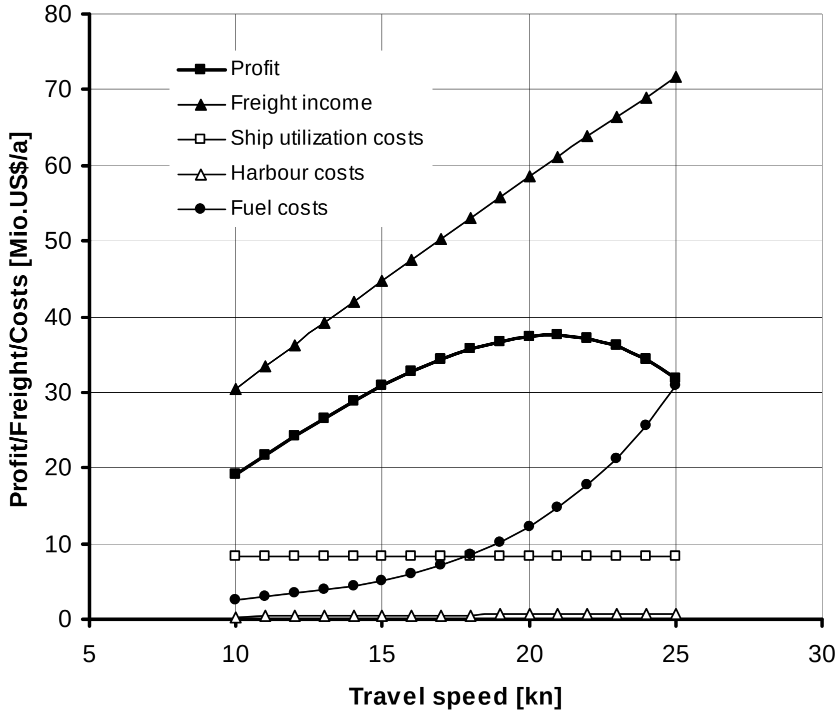


Figure 3. Speed dependency of the operating profit for full utilization of the effective capacity

3.2 Profit-optimal Speed

The determination of the profit optimal speed considers the average freight rates, the effective ship capacity, the number of harbour stops, harbour prices and the average bunker prices times the ascent parameters of the fuel consumption curve.

Our calculations show that the profit-optimal speed decreases inverse proportional to the n-root of the bunker price, slightly faster than the cost-optimal speed. It is also independent from the ship utilization price and increases with the effective ship capacity. Thus it is higher for large ships than for smaller ships as long as their capacity is fully used.

Profit optimal speed also decreases with decreasing freight rate and reaches the cost-optimal speed if the achieved freight rate falls to the freight cost rate at full capacity utilization. It decreases only slightly with increasing number of harbour stops and longer harbour times and as far as the achievable freight rates do not depend on the distance, the profit-optimal speed is independent of the route length.

3.3 Comparison and Consequences

The model calculations were performed with a basic MS-Excel-version of a fleet planning tool developed by the authors. This spread sheet program consists of input-tables, such as Table 1 and Table 2, output-tables, such as Table 3, and backup sheets, where the different calculations are executed, and tables, where intermediate data are stored.

Table 3 presents the result of three different scenarios for serving an outward and inward freight demand of more than 260,000 TEU/a between Rotterdam and Shanghai by a fleet of 5000-TEU-container ships. The ship data are taken from Table 1, the operating data from Table 2.

Table 1. Key data of a container ship used for the model calculations[2]

Key data	Characterization
Type	Panamax Containership
Load Units	20 ft Container = 1 TEU 40 ft Container = 2 TEU
Maximal capacity	5,000 TEU
Ship Utilization Price P_S	23,000 US-$/d
Design speed	v_{min} = 12.5 kn v_{max} = 25.0 kn
Fuel consumption at v_{max}	320 kg/sm
Consumption rate $c_F(v)$	$c_F(v) = 58 + 0.00013 \cdot v^{4.5}$

Table 2. Scheduling and operating data for the round tour of a container ship[3]

	Outward tour	Return tour
Driving Length L_i (sm)	11,000	11,000
Bunker Price P_B (US$/t)	500	500
Freight rate P_F (US$/TEU)	1,050	630
Harbor stops N_H (per tour)	1	1
Stop time t_H (h/stop)	48	48
Stop price P_H (US$/stop)	42,000	42,000
Filling degree ρ_{max}	95 %	95 %
Freight Limit Performance μ (TEU/year)	42,633	42,633

The first column of Table 3 contains the results for a fleet of 6 ships travelling with the maximal design speed of 25.0 kn, the second column for a fleet of 7 ships travelling with the profit-optimal speed of 20.7 kn and the third column for a fleet of 9 ships travelling with the cost-optimal speed of 15.3 kn. In all three cases, the demand is served weekly with a total freight limit performance of the fleet around 255,000 TEU/a. Amazing are the resulting differences between operating costs, profits, fuel consumption and emissions.

Our calculations show that despite the additional investment and the lower freight rates which have been assumed for compensating the longer transport times, the profits are increased by 28 % or 52 Mio.US$ when travelling with 7 ships at profit-optimal speed, and increased by 34 % or 63 Mio.US$/a when travelling with 9 ships at cost-optimal speed.

For the business case of Table 3, the reduction by travelling with profit-optimal speed is 45 % and saves annually 164,000 t fuel, 550,000 t CO_2 and 16,000 t SO_x, and with cost-optimal speed 70 % and saves annually 261.000 t fuel, 818.000 t CO_2 and 24,000 t SO_x.

Applying these saving percentages to the present global maritime fuel consumption and emissions of carbon dioxide, sulphur oxides, nitrogen oxides and soot, demonstrates the utmost importance of slow steaming in comparison to other means and measures which are tried or discussed presently in order to save fuel and reduce emissions (Bond 2008; Corbett et.al. 2009; Faber et al. 2010; Meyer 2011).

[2] Ship utilization price: Cost-rate for own ships or charter-rate for third-party ships
[3] Based on a roundtrip with only two stops in Rotterdam and Shanghai.

Table 3. Fleet planning results for three scenarios with maximal speed, profit-optimal speed and cost-optimal speed and adapted number of ships[4]

	Maximal design speed	Profit-optimal travel speed	Cost-optimal travel speed
Tour length (sm)	22,000	22,000	22,000
Service frequency per year	54	53 (-2 %)	57 (+ 6 %)
Travel speed (kn)	25.0	20.7 (-17 %)	15.3 (-39 %)
Transport time (days/dest.)	20	24 (+19 %)	32 (+57 %)
Fleet size (ships)	6	7 (+17 %)	10 (+ 67 %)
Fleet performance (TEU/a)	255,799	251,354 (-2 %)	271,268 (+6 %)
Freight rates outward (US$/TEU)	1,050	1,000 (-5 %)	950 (-10 %)
Freight rates return (US$/TEU)	630	600 (-5 %)	570 (-10 %)
Freight revenues (Mio. US$/a)	430	402 (-6 %)	412 (- 4 %)
Operating costs (Mio US$/a)	934	634 (-32 %)	522 (-44 %)
Operating profits (Mio US$/a)	191	243 (+27 %)	271 (42 %)
Fuel consumption (t/a)	369,532	193,945 (-48 %)	107,880 (-71 %)
Fuel consumption (t/a)	1.4	0.8 (-47 %)	0.4 (-72 %)
Carbon Dioxide (CO2, t/a)	1,156,636	607,047 (-48 %)	337,665 (-71 %)
Sulfur Oxydes (SOx, t/a)	33,258	17,455 (- 48 %)	9,709 (-71 %)

4 CONCLUSION AND OUTLOOK

Out of these results, we can see that cost-optimal and profit-optimal speeds can be used for the determination of optimal business strategies. Large operators following a cost-leader expansion strategy can reduce during times of decreasing freight demand and low freight rates the travel speed of their fleets stepwise down to a cost-optimal value.

Limitations of the study refer to the narrow understanding of environmental sustainability due to reduction of GHG emissions that is mainly due to reduction of bunker fuel due to a lower speed. Other issues of environmental sustainability have been excluded in our discussion.

By using the achieved cost savings, lower freight rates can be offered in order to attract additional freight and to ensure sufficient utilization. In times of increasing freight demand and high freight rates on the markets, the travel speed is increased up to the profit-optimal speed. The freight rates can be increased as long as the utilization is not affected. However, business practice has shown that the main motive for slow steaming are reduced costs.

REFERENCES

Bond P. (2008), *Improving Fuel Efficiency through Supply Chain? And the Ship Management Plan*, http://www.intermanager.org/2008/11/improving-fuel-efficiency-through-the-supply-chain/.

Cariou, P. (2011): Is Slow Steaming a Sustainable Means of Reducing CO_2 Emissions from Container Shipping?, *Transportation Research,* **16D**, 260-264.

Corbett J. J., Wang, H. and Winebrake, J. J. (2009): The Effectiveness of Speed Reductions on Emissions from International Shipping; *Transport Research*, **14D**, 593-598.

Eyring, V., Isaksen, I.S.A., Berntsen, T., Collins, W., Corbett, J.J., Endresen, O., Grainger, R. G., Moldanova, J., Schlager, H. and Stevenson, D. S. (2010): Transport Impacts on Atmosphere and Climate: Shipping, *Atmospheric Environment*, **44**, 4735-4771.

Faber, J., Freund, M., Köpke, M. and Nelissen, D. (2010): *Going Slow to Reduce Emissions. Can the Current Surpluse of Maritime Transport Capacity be Turned Into an Opportunity to Reduce GHG Emissions?*, Seas at Risk, http://www.seas-at-risk.org/pdfs/speed%20study_Final%20version_SS.pdf.

[4] Ship data from Table 1, operating data from Table 2.

Gudehus H. (1963): Über die optimale Geschwindigkeit von Handelsschiffen [on the optimal speed of cargo ships], Hansa, Hamburg, **23**, pp. 2387

Gudehus H. (1967): Über den Einfluß der Frachtraten auf die optimale Schiffsgeschwindigkeit [on the correlation of freight rates and the optimal travel speed of ships], Schiff&Hafen, **3**, pp. 173.

Gudehus T. (2010): Slow Steaming Schiff &Hafen, **4**, pp. 12; **6**, pp. 12.

Gudehus, T. and Kotzab, H. (2012): *Comprehensive Logistics*, 2nd revised and enlarged edition, Springer, Heidelberg et al.

HSH Nordbank (2009): *Betriebskostenstudie 2009. Untersuchung der Betriebskosten deutscher Containerschiffe – Kurzfassung* [Study on operation costs 2009. Examining the operation costs of German container ships – Short version], Hamburg.

Meyer J. (2011): *Slow Steaming in der Contanerschifffahrt*, TU Hamburg-Harburg

Notteboom, T. and Vernimmen, B. (2008): The Effect of High Fuel Cost on Liner Service Configuration in Container Shipping, *Journal of Transport Geography*, **17**. 325-337.

Psaraftis, H. and Kontovas, C. (2010): Balancing the Economic And Environmental Performance of Maritime Transport, *Transportation Research*, **15D,** 458-462.

Ronen D. (1982): The Effect of Oil Price on the Optimal Speed of Ships, *The Journal of the Operations Research Society,33*, 1034-1040-

Schneekuth H. and Bertram K. (1998): *Ship Design for efficiency and Economy, Butterworth-Heinemann*, 2nd ed., London/UK.

Song, D.-P. and Xu, J. (2012): An Operational Activity-Based Method to Estimate Emissions from Container Shipping Considering Empty Container Repositing, *Transportation Research Part D*, **17**, 91-96

RAIL LANDSIDE CONTAINER TRANSPORT INTERFACES

Hans G. Unseld

Unseld, H .G., CargoTechnologies GmbH, A-1080 Vienna, office@raiload.at

ABSTRACT

This paper takes a fresh look at physical container handling from a process perspective for two reasons: when introducing automation in container handling, dynamic performance data need to be specified and the lack of that data at present makes it difficult to compare new developments with existing terminal designs and performance data on a common ground. For this reason, a modified process model for rail-inclusive interfaces is suggested along with a new characterisation data set especially for the inclusion of up-to-date criteria. The verification and validity process have started and first results look promising. This paper deals also with a particular challenge for new terminal technology development. The market environment is very competitive, personal networks play an important role and background decisions are driven by political considerations at a high extent. In this challenging context, competition means here: established tradition and scepticism of a technology-critical transport and logistics clientele. Solid and consistent data and transparency of communication and messages are needed. The new characterisation data set and the framework suggested will help to define common grounds for implementing new technologies in rail transport on the one hand, and also assist actors to better communicate between the different playing field on their strategies and needs on the other hand. This aspect is of great importance when defining future rail transport network reference designs in the context of TEN-T pilot installation for main, intermediate and small nodes.

Keywords: Intermodal terminal, rail networks, physical handling, interface performance, container, material flow dynamics

1 INTRODUCTION

Managing physical interfaces are among the trickiest challenges in systems technology. They deal with moving objects and the way how they are separated and reintegrated with their different carriers without any distortion of their physical integrity. Successful systems are equipped with efficient interfaces as obvious essentials in order to allow streams flowing effectively through networks. They are the real core pieces for any network. Examples range from distribution networks for water, gas and electricity to more complex ones like communication networks, which are the backbone of today's communication and IT systems. This paper aims at presenting the scientific community a new technology interface solution and a new definition, along with a new set of criteria as an input for a fresh debate on the commercial and societal potential of rail transport networks for hinterland transport in Europe.

This paper deals with interfaces in rail container transport and their actual potential for supporting future mass transport rail networks. The paper is built upon analyses of state-of-the-art interface designs and conclusions drawn; it suggests a set of dynamic network specific parameters for comparing different approaches. These data sets will then serve as input for future research on potentials of new interfaces, new terminals, new transport networks and new logistic services in terms of impact on value added logistics services for the transport actors and for the prosperity of surrounding regions in particular. The paper concludes with thoughts for the design of a novel type of terminal.

2 CHARACTERIZATON AND EVOLUTION OF TRANSPORT INTERFACES

Interfaces for the physical movement of a container flow between different modes of transport in an interface context are simple in function, but highly interdependent on complex processes. Their main behaviour may be characterized as an "external event triggered predetermined function for moving a container". Being part of a node, any interface interconnects physically the container transport carriers with internal processes, like storing, sorting, turning and other transport or logistics related services. Figure 1 explains the process model: in an intermodal node or terminal, two interfaces maintain the physical flow of containers between the "external" or public transport modes and two further interfaces interconnect the storage with the intra-terminal transport system (red box line).

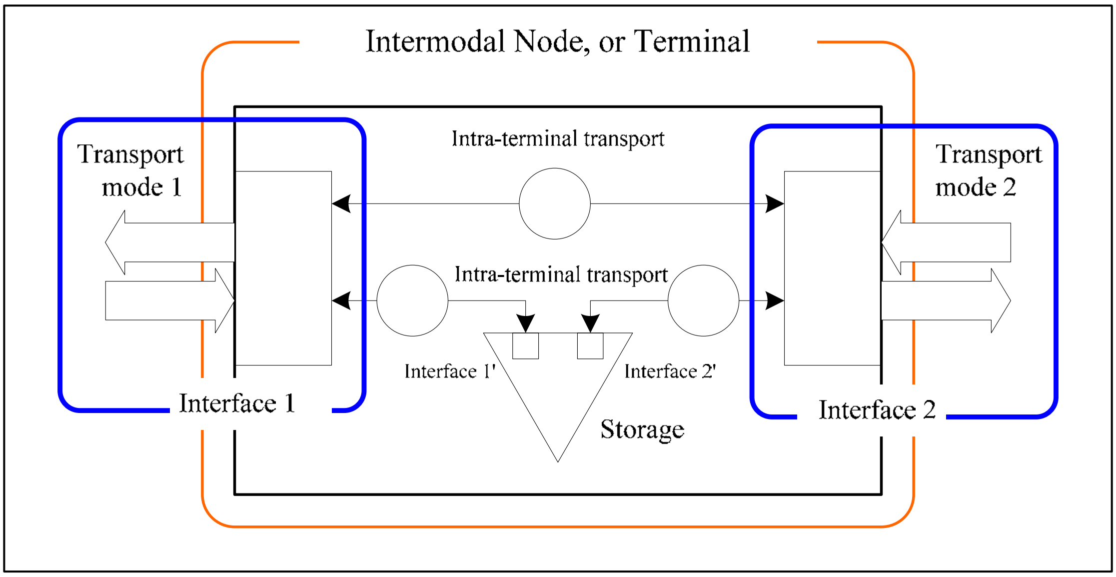


Figure 1. Process Model for Interfaces as Part of an Intermodal Terminal or Node

Moreover, interfaces in rail transport systems are determined by rail infrastructures as well as by site-specific conditions and requirements and therefore serve often as kernel or core process of intermodal terminals and rail loading facilities. These processes, their design and performance determine the function of this particular terminal: as a nodal point for either dedicated train services or, in future, as a node within a transport network. The nodal performance depends to a high extent on the process engineering and the degree of integrating interfaces with the nodal services. In this context, a future network can be interpreted as well as a service operation resource universe with a portal for any freight actor serving the consignee as final user of the transport chain. In this perspective, a direct interdependency does exist between the interfaces' physical process performance and the features of the whole transport network being part of it. This multi level approach is subject of further research and results will be reported accordingly, including the interrelation between the physical handling interface and the characteristics of a network and the performance.

Container transport by rail today is performed between two nodes and tracks interconnecting them. Their main purpose is to enable container transport by rail as main haul from location A to location B via nodal points close by. In future and with significant higher transport volume, their function will be upgraded towards maintaining and managing a more continuous and time-controlled physical flow of units packetised according to logistics and cost criteria within a competitive environment. In this type of high volume container transport network to come, interfaces have to comply with additional network specific parameters.

Today and tomorrow, interfaces enable and control that transport and flow as part of nodes' architecture and function. The art of integrating all elements according to the conditions of an efficient management of physical flows on a particular site makes the interfaces indispensable, yet unique parts of any network. In this context, new ways of designing lay outs for interfaces and nodes will help to upgrade existing terminals, and support deploying additional new brown field and green field constructions. New characterisation criteria should lead the way.

2.1 Proposal for Characterization of Interfaces

Interfaces for physical container handling are characterized by their ability to match loading performance with the transport characteristic of the carrier associated with (see Fig. 1, blue lines). Two areas were identified as being critical: while today terminal designs are characterized by static data, the network requires mainly dynamic data and - since in the past terminal and transport operation were kept separately - it is suggested to define a comparative data set for the interfaces which must involve all related interface actors. This set of data could well serve a basic data source, for example even for a service level arrangement framework. It is suggested to split the characterization of future interfaces into six dimensions, as indicated in table 1.

Table 1. Characterisation of Interfaces for Physical Handling of Containers

Interface dimension	Characterisation	Dimension
Interface key performance data	Throughput from one mode to another mode Speed of throughput (typical and maximum value)	no. of Boxes / day no. of Boxes / hr
Interface efficiency Net time Total time	Coefficient of value adding times of loading plus unloading vs. the average transport time of the respective carrier - operation times only - including preparation and operation time	 % %
Interface agility	Reaction time for performing and concluding a reference interface operation, related to the total transhipment time	%
Interface energy efficiency	Electric energy equivalent for performing a complete Interface operation with reference container (40 tons)	kWh / lift
Interface layout flexibility and land use	Value added by interfaces and other functions or by total node vs. operating area or total land use	€ / square meter
Interface cost effectiveness	Costs of interfacing reference interface operations, with 50% utilisation of installed capacity per day	€ / lift per day

The interface key performance data are classical material flow data: throughput measured without restriction at both sides of the interface over a given period of time. This set of data is part of the design parameters for erecting the terminal and cannot be modified in an existing terminal without any investments. The other parameters are dynamic or volume-related data, very much needed for characterization of interfaces in nodes as part of an agile transport network.

The interface efficiency is an expression of the degree of value adding match between the carriers in terms of percentage needed for interrupting its key activity: value adding transport services. Base line for calculations is service hours. For high throughput interfaces with transport network capabilities, loading times must be short in relation to the average transport time.

Interface agility determines the response of a call for action of the interface in terms of minutes per standardised test case per carrier type. Mainly a matter of planning is a short reaction time a key ingredient of a networks flexibility to react on interrupts and disruption.

Energy efficiency determines the electric energy equivalent needed for a standardised lift operation. In "greening transport" strategies, energy saving operation with heavy goods bears large potentials for savings. It has four dimensions, namely energy consumption, speed of operation, and environmental criteria and operating costs. A further dimension in correlation to the ones mentioned above is layout and area occupation for the performance of value adding processes in comparison with the cross area occupied; this parameter needs further research and is not considered essential for this paper.

More throughputs in nodes mean more operational area according to existing terminal design rules. Therefore, an important strategic issue is to encounter ways of designing flexible layouts for nodes, i.e. within the existing rail network infrastructure as brown field development. A new criteria is suggested which honours a careful land use in relation to value adding service. A calculation method is suggested which will result in values added per annum in € per square meter of operating area or total layout (= land use) area.

The interface cost effectiveness describes how given interface investment impacts the total cumulative operational costs of all actors involved in a NTR (near-to-reality) test scenario. This scenario accounts for the sum of all value adding handling operations during an operational day; base line is a site specific typical reference day and a given (i.e. 50%) average system utilisation of the interface operation. With this approach a methodology should be introduced which could become a figure-of-merit for interfaces in their complete function, and not only for a single part of it.

2.2 Interface Efficiency and Agility

In order to describe and compare interfaces' performance data in addition to static key performance parameters, a dynamic interface efficiency and interface agility measure is suggested to be introduced. The dynamic performance of a physical container handling interface is primarily defined by the carriers' interrupts that cause unproductive time at interfaces. The first indicator suggested relates the value adding time of loading plus unloading vs. the transport time of the respective carrier. Two figures are important: 1: net efficiency measure, including value adding time only and 2: total efficiency measure, which includes mooring, shunting and gate passing operations for both actors. While interface efficiency describes the losses caused by non-productive times, the interface agility outlines the reaction time in unified selective cases. Examples are: the time for calling minimum 100 empties or similar cases which may causes disturbances at the carriers end and terminals with high performing interfaces can react.

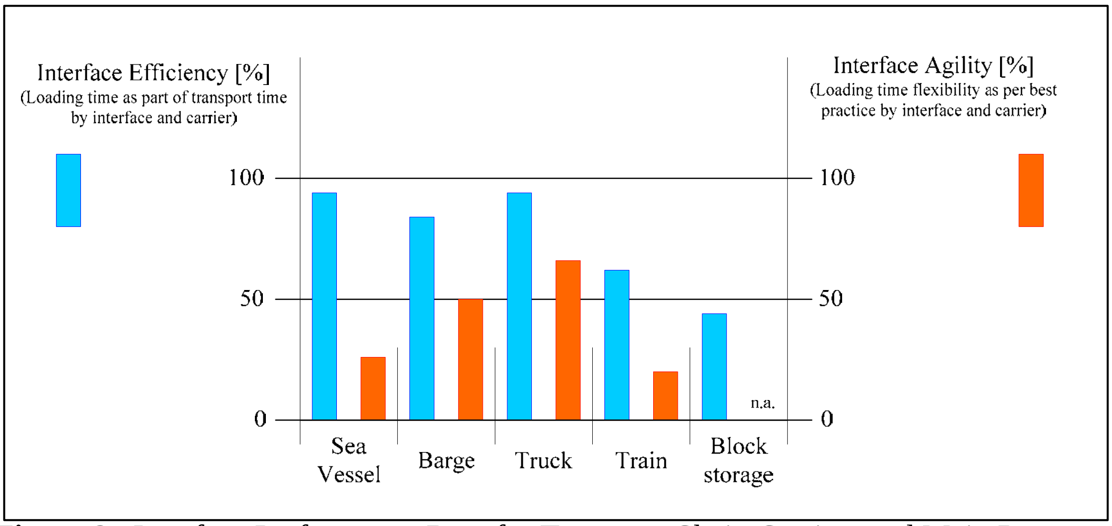


Figure 2. Interface Performance Data for Transport Chain Carriers and Main Resources

In Fig. 2, both the interface efficiency and the agility measures are shown as indication for main resources in container transport. The exceptional dominance of the today's work horses

serving container- and hinterland transport - sea vessel and truck – is shown here. The low interface performance for rail transport is an average value, and specific shuttle services show better results. The long loading interface time in rail interfaces has been identified as main hurdle for not supporting rail network functionality. One even can say: rail transport with this type of low performance is unable to undertake any role within a rail transport network.
Further studies will be carried out in order to validate the results in more specific cases for which feasibility studies are currently under way.

2.3 Measures to Improve Throughput

The static performance of a physical container handling interface is characterized by throughput volume and speed of throughput as typical and peak data. It is established by applying unrestricted flow conditions at both sides of the interface. Interfaces' throughput, or container movement per time, depends on equipment, layout and material flow design and node management. At present, the 28 German KV-Terminals average a flow factor of less than 2, which accounts for less than two trains per day per loading track, or less than 400 TEU per day maximum.[1] Counting average numbers, the daily quantity is actually at a level of less than 200 TEU per terminal!

Despite this low throughput level with actual terminals, future transport demands for container on rails will raise and the question remains: where is the limit of throughput at rail interfaces and what kind of rationalisation is feasible and recommendable for when? The following five different measures for increasing the throughput have been selected after validation[2] [3]:

1. Reducing preparation and idle time, i.e. for wagon grouping operation with incoming and departing trains
2. Speeding up the loading process with faster equipment and by parallelisation of it
3. Concentrating the rail loading operation along, or at least close to existing rail lines within rail infrastructures for material flow reasons
4. Process automation to allow remote controlled 24/7/365 hrs operation at the interfaces and intra-terminal transport services
5. Integration of buffer stocks into the rail loading or rail interfacing processes.

All these measures represent a large potential for throughput enhancement at interface level. They are subject of further debate in chapter 3 of this paper. However, there is some evidence that new designed interfaces allowing automated, remote-controlled, parallel services at trains while they perform an operational halt on a loading track will deliver significant higher performance levels compared to loading systems in existing terminals.

2.4 Challenges Resulting from this Interface Characterization

The steadily growing container transport volume urgently calls for more innovation in comodal interfaces including transhipment technologies[4] in order to increase utilisation of existing rail infrastructures. The challenges include: [5] "Automatic train operation and control will allow more traffic on existing track, but transport hubs and other infrastructure will need radical redesign to cope with the doubling of traffic in situations where there is little or no available land for expansion of stations and freight handling facilities…"

[1] UIC, DIOMIS (2007a)
[2] Unseld, H.G., Kotzab, H. (2011)
[3] UIC, DIOMIS (2007b)
[4] Olson, B. (2012)
[5] Amoore, J. (2012)

From these preliminary results and from feasibility studies done so far, the technology concept presented by the authors in 2011[6] shows a very high potential to fulfil the challenges stipulated. Steps for implementation are presented in the conclusion chapter.

3 CONCLUSION

Sub title: Thoughts when Designing a New Rail Network Node.
Rail transport is associated with many challenges: heavy investments, the loss of public acceptance for noisy transports, low logistics performance, etc. We addressed the question: How can technology support rail container transport in making these attributes obsolete?

That offers chance no. 1. Container transport is a profitable transport market with lots of opportunities for growth. This means for IM's (Infrastructure manager): installing new node and altering existing terminals according to new designs which carry the sign: "Yes, we do mean what we say - environmentally friendly transport technology". It would mean for RU's (Rail undertakings): "We are proud actors for the energy turn around". Ample opportunities exist in Europe. Some of them are presently under consideration. How can we proof our concept? Should we build a demonstrator again?

That offers chance no. 2. The characterisation framework suggested in this paper is a result of lengthy uneven debates about design principles, doubts about new technology, costs and benefits of present terminal designs which will be lost, when introducing innovations for maintaining and may win market shares against truck transport. Innovation means: a significant step forward operation with new ideas in line with the raiload loading concept. Within the coming months, at least one example will be compared and validated; present data points towards proof the concept. This leads now back to the question of planning terminals and a concise data framework.

That offers chance no. 3. The legislation for design, construction and operating intermodal terminals is a result of a long time process. The flexibility for new innovation is close to zero. Here again, the new characterisation framework should set the scene for next coming debates. The present trend with limited funds to invest in infrastructure, a fresh look is needed and as a consequence regulation must be review, based upon clear data. They must be understandable, communicable and must reflect future needed dynamic systems behaviour for rail transport of containers in rail based co-modal networks.

The framework suggested here has been elaborated to become a standard characterisation data set for transport network designers, terminal designers, and for developers of novel rail transport products when dealing with future rail transport networks.

REFERENCES

Amoore, J. (2012), ERRAC Work Package 05: Strengthening Competitiveness, p. 21, In *Competitiveness Roadmap*, Brussels (to be published in mid 2012).

Kotzab, H. and Unseld, H. G. (2012), Zukunftsweisende Konzepte des containerisierten Seehafenhinterlandverkehrs auf der Schiene, (submitted for publication in mid 2012).

Olson, B. (2012), ERRAC Work Package 02: Encouraging modal shift (long distance) and decongesting transport corridors, p. 36, In *Freight Roadmap* (to be published in mid 2012).

Unseld, H. G., and Kotzab, H. (2011), The potential of High-Performance Transports in Rail Networks – The case of Advanced Extended Gateways and Rail Network Nodes for Rail-Based Hinterland Container Traffics, In T. Blecker, C. Jahn and W. Kersten (eds), *Maritime Logistics in the Global Economy*, p347-366, Eul Verlag, Lohmar.

UIC, (2007a), DIOMIS Best practices for management of combined transport terminals, Paris

UIC, (2007b), DIOMIS Final report WP A5: Improving the use of available train length, Paris

[6] Kotzab, H. Unseld, H. G. (2012)

A STUDY ON M2M BASED CONTAINER TRACKING SYSTEM FRAMEWORK FOR GLOBAL SCM

Hyung Rim Choi[1], Kang Bae Lee[2], Joong Jo Shin[2], Jung Rock Shon[1], Jin Wook Lee[2*] and Hee Mok Son[2]

[1]*Department of Port & Logistics Systems, Dong-A University, Busan, Korea*

[2]*Department of Management Information Systems, Dong-A University, Busan, Korea*

ABSTRACT

Logistics paradigm is changing from logistics cost cutting to safe transportation and requiring diverse information services. And transportation volume of global container is increasing as trade grows larger than before based on lifting trade barriers. At the same time, participants using logistics service require real-time based information of container location, security for preventing a robbery case, and temperature/humidity/impact. But existing system based on bar code, EDI, RFID technology is possible to communicate only part of point in all logistics network. In addition, constructing infrastructure cost high because logistics network is based on all over the world these days. Container tracking system that is suggested in this research is one of the methods for realizing Smart SCM(Supply Chain Management). This would provide feature for managing container as communicating data between tracking device and information gathering server based on real-time. Moreover, suggested prototype system is applied to TSR(Trans-Siberian Railroad) which has a length of 9,288Km in Russia, and then application result is analysed. In the result of analysing the data collected in the case of TSR transportation, from the viewpoint of Smart SCM, it was possible to find out whether the container transportation is punctual and whether the cargoes are damaged or not. In addition, from the viewpoint of constructing and operating the container tracking system, considering the efficiency of the battery use of real-time container tracking devices, it was necessary to select a mobile communication company in accordance to the country which the container is transported to.

Keywords: Container tracking system, Global SCM, Real-time Container tracking device

1 INTRODUCTION

In the era of limitless competition, as the management surroundings of each enterprise have become worse, global supply chain management has drawn attention. The reason is that production costs should be lowered through global sourcing and the counterforce to market signals should be raised. As though establishment of those global supply chain management systems is important core competences of businesses, supply chain managements have not been managed properly except several major companies due to the limitation of the methods to manage supply chains and the previous technologies (2012).

In addition, the requirements of the customers that transport their goods using containers have been changed from the cost saving aspect into the service aspect to make sure whether the cargoes are transported punctually such as the safe transport of cargoes, the state/position tracking of cargoes, prompt customs clearance, and so on. However, as the previous RFID-

[*] Corresponding author: jw6416@gmail.com

based container management system is difficult to provide container sealing records from shippers to consignees and has the limitation to provide real-time base passing information and location tracking information, it is necessary to provide the information that users want by applying the M2M(Machine to Machine) based real-time mobile communication technology whose applicability has been increasing.

Especially, in applying the M2M based real-time mobile communication technology and so on, the systematic communication standard and data format between the real-time container tracking equipments and the server collecting information are required for efficient information transmission and reception, but analysing the recent researches, the researches on the data transmission/reception system that can be applicable in various communication environments have not been developed.

This research is to suggest the communication method having systematic communication standard and data format in communicating between real-time container tracking device(ConTracer) and information collecting server for container tracking system built for realizing smart & green SCM.

2 LITERATURE REVIEW

The advanced researches related to container tracking system for realizing smart & green SCM are as below. Yingli Wang (2008) have studied on the real-time communication technology that can be applied to container transportation, and tried to suggest the possibility to apply the technology through the comparison of strengths and weaknesses between RFID technology and telemetric technology. In Smart Container Chain Management (Smart-CM) project, Julia Carn (2011) pursued the case study on tracking the state information of container transportation in real time, but did not suggest a specific management and communication method. Su Jin Kim (2008) proposed Intelligent Networked Containers as a method to realize global SCM, but it was based on RFID technology and has the limitation that containers were managed through the data collected only in main logistics bases, not real-time communication in all the way to the destinations.

Chae Seok Lee et al. (2010) researched into designing and realizing the transport protocol in large tag data on the system based on RFID, and Woo Sik Bae et al. (2009) researched into efficient authentication protocol by making use of eSeal which is container security equipment based on RFID. Besides, Jeong Tae Kim(2011) made the study of light-weight protocol for tag security in RFID system, and Hyung Rim Choi et al. (2006) suggested the standard for data communication in applying RFID-based container management system to port logistics industry. The above previous researches used RFID-based container tag and information system for the purpose of managing containers in logistics industry, but there was the limit that those researched dealt with the communication methods and protocols in RFID-based position.

Besides, in the M2M-based environment transmitting/receiving real-time communication and sensor information, not the RFID-based system, Lee and Hong (2010) developed the message protocol supporting the interoperability among application systems by reflecting the feature of standardized M2M-based sensor system, and It means that the research for improving communication performances has been done, but it has differences from the protocol to transmit/receive the information necessary for logistics industry.

Anuparp Boonsongsrikul et al. (2011) researched into false data recognition protocol for performing minimum communication cost in wireless sensor network, and Yoonbeom Choi et al. (2011) researched into the optimization of stationing energy supply equipments for decreasing the operating expenses of sensor network. It could decrease communication loads, but it differs from the research related to the direct cost saving of mobile communication. Young-Hye Min (2011) have conducted the research for decreasing communication costs by suggesting the optimal compaction technique selection method for the rapid effective transmission/sharing of data files, but there is a difference in that the researched was conducted in the normal wired and wireless internet environment, not in the wireless network environment using mobile communication module. Woo Sik Bae et al. (2011) suggested the authentication protocol that satisfied security standard required in RFID system and minimized communication costs by minimizing of the amount of arithmetic operations such as random number generation, etc., but it differs from this research in that it is RFID system not using real-time mobile communication module.

3 CONTAINER TRACKING SYSTEM

The real-time container tracking system for smart & green SCM introduced in this research is attached to the inside of the container carrying the real-time container tracking device(ConTracer) as described in Fig. 1, Fig. 2, and Fig. 3, it detects the information of location and status (temperature, humidity, vibration and the opening and shutting of the door) of the container carrying the equipment in real time. The information detected from the real-time container tracking device is transferred to the middleware and monitoring system through communication infrastructure such as mobile communication base station, etc., and as users can make sure the information such as the location of the container and the temperature, humidity, vibration and door opening/shutting of the inside of the container, they are able to make proper decisions for global SCM.

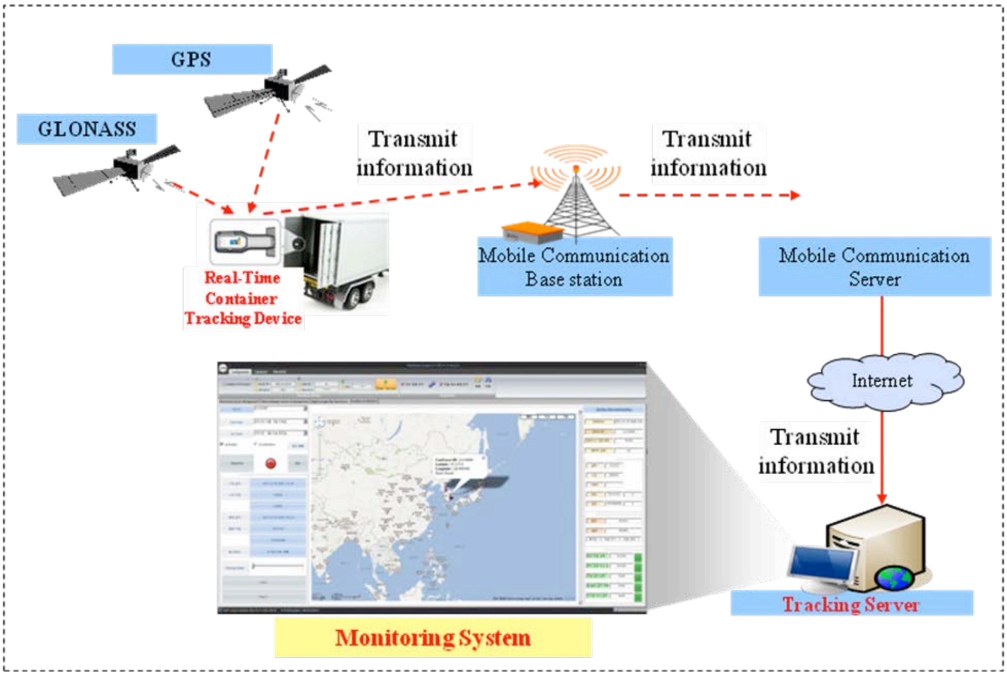


Figure 1. Real-time Container Tracking System for Smart & Green SCM

The real-time container tracking system for smart & green SCM introduced in this research is attached to the inside of the container carrying the real-time container tracking device(ConTracer) as described in Fig. 1, Fig. 2 and Fig. 3 and detects the information of location and status (temperature, humidity, vibration and the opening and shutting of the door) of the container carrying the equipment in real time. The information detected from the real-time container tracking device is transferred to the middleware and monitoring system through communication infrastructure such as mobile communication base station, etc., and as users can make sure the information such as the location of the container and the temperature, humidity, vibration and door opening/shutting of the inside of the container, they are able to make proper decisions for global SCM.

Figure 2. Real-time Container Tracking Device(ConTracer)

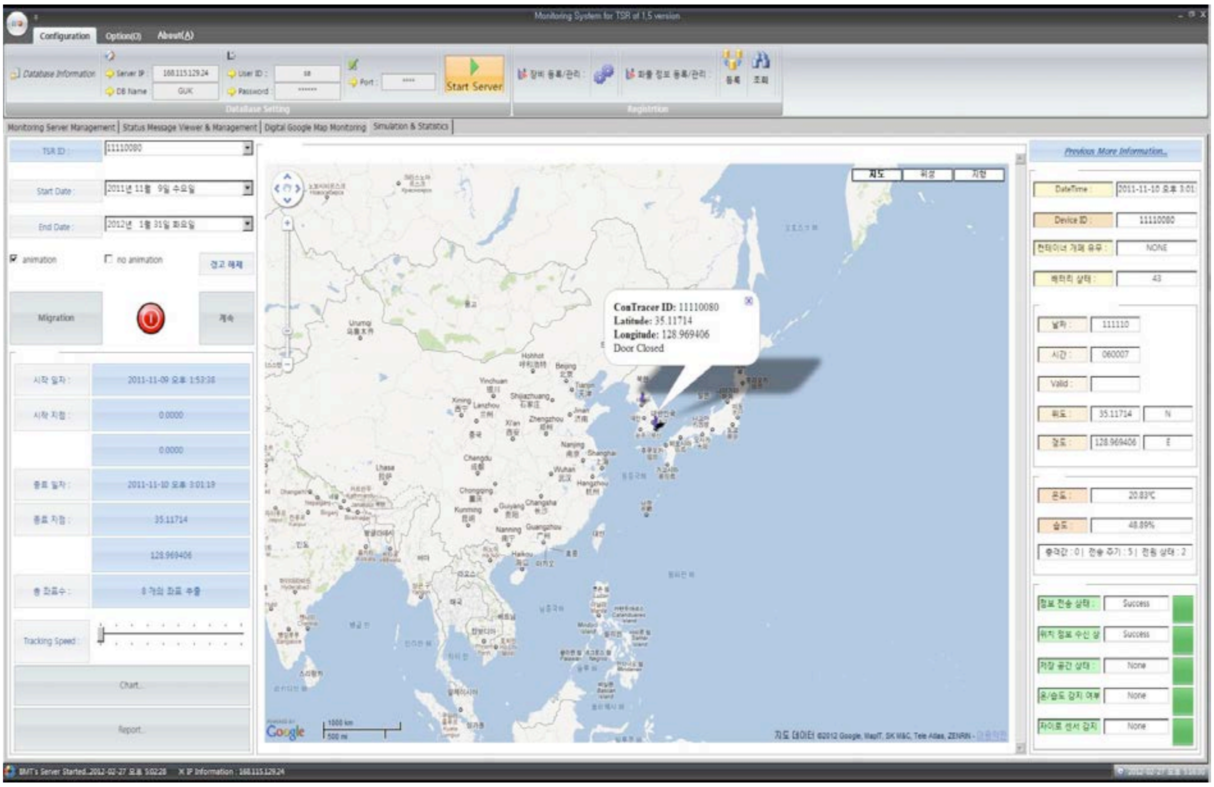


Figure 3. Real-time Container Tracking System(Middleware + Monitoring System)

Fig. 4 shows the organization of the container tracking system which the data communication method with the systematic communication standard and data format in the communication between the real-time container tracking device and the location information collecting server.

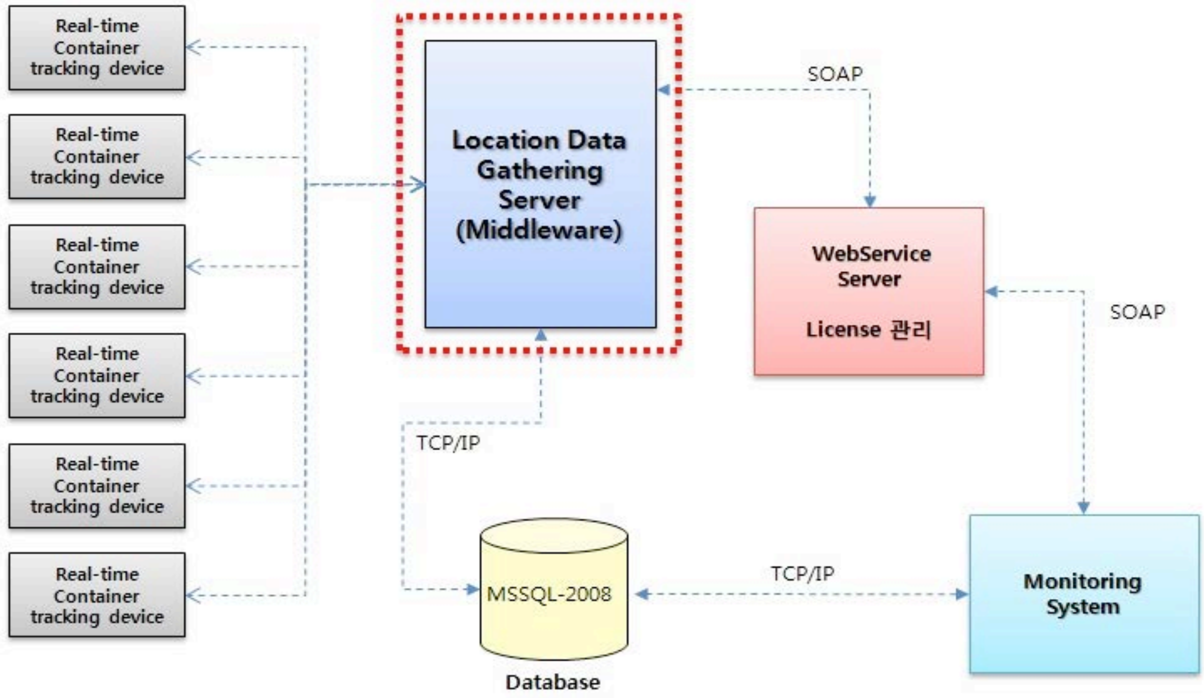


Figure 4. Component block diagram of Real-time container tracking system

The real-time container tracking system consists of the real-time container tracking equipments for making sure the location and state(temperature, humidity, impact and door opening/closing) information of the container, the location information collecting server that collects the location and state information from the real-time container tracking device, receives an order from the broadcast server and transfers the order to the relevant device, the database for storing the location information, the web service server for the license management and web service, and the monitoring system for expressing the collected information.

In container tracking system, the communication technologies applied to the real-time container tracking device are GSM(Global System for Mobile communications) and WCDMA(Wideband Code Division Multiple Access) methods, and in the identification of location information, the location and time information of the device is acquired by using GLONASS(GLObal NAvigation Satellite System). Of course, according to applied environments and objects, it is not limited to the above communication methods and radio wave satellite navigation system, and other methods and navigation systems can be applied.

In container tracking system, the state identification technology applied to the real-time container tracking device is composed as follows: In the interior of the real-time container tracking device, the temperature/humidity/impact sensors are applied and they detect the interior environments of the container, and the reed sensor makes sure the status of the door opening/closing of the container.

The container tracking device of the container tracking system makes sure the time using RTC(Real Time Clock), and in case of not obtaining the time information of GPS or GLONASS, the time transmitted earlier is used. In addition, the time information is used as write time in recording the activate log of the real-time container tracking device.

In addition, the real-time container tracking device synchronizes time by using UTC(Universal Time Coordinated). To do that, when the electric power of the device is initially operated, it uses the UTC value that is acquired through the communication with the mobile communication base station, and synchronizes the time of RTC by using GNSS(Global Navigation Satellite System).

As the above diagram, in transmitting data from the real-time container tracking device, SEED(SEED Block Encryption Algorithm) encryption is performed, and in receiving the order from the location information collection server, SEED decoding is performed, and according to the operational environments and user's requirements, the encryption of AES(Advance Encryption Standard) and DES(Data Encryption Standard) can be selected and encryption or decoding is possible.

4 INFORMATION TRANSMISSION/RECEPTION METHOD OF REAL-TIME CONTAINER TRACKING SYSTEM

4.1 Information Transmission Method

The method to transmit the container information from the real-time container tracking device to the server is as follows. The real-time container tracking device transmits information according to the established cycle (basically 1 hour), and at this time, using TCP/IP socket communication, it transmits information to the server by the below payload(40 bytes). In transmitting information, SEED encryption algorithm is used for the information protection of payload 40 bytes, and in case of not transmitting the information to the server due to the surroundings(ships, steel-frame structure or no mobile communication base station), one more retransmission of the information is supposed to be tried. When information is transmitted to the server in this real-time container tracking device, the detailed explanation of Payload is in the following table.

Table 1. Payload of Information Transmission in Real-time Container Tracking Device

Section	Definition	Length
Index	Log index stored in the real-time location tracking device	2byte
Protocol ID	ID of the relevant protocol. The ID of this protocol is 0x01.	1byte
ConTracer ID	Unique number identifying ConTracer	8byte
Date & Time	Indicating the time of the event occurrence and information transmission of the device	6byte
Location Data	Indicating the current location of the container on the basis of GPRMC data of GPS	11byte
Temp	Temperature value measured in the real-time location tracking device	2byte
Humid	Humidity value measured in the real-time location tracking device	2byte
Impact	Expressing the vector sum of the acceleration value of X, Y and Z axes of the 3-axis acceleration sensor.	2byte
Container Door Status	Indicating the status of container door (shutting/opening)	1byte
Battery Capacity	Indicating the percentage(%) of battery residual capacity	1byte
On/Off Status	Indicating the status of initial On, In Progress and Off	1byte
Transmission Cycle	Indicating the cycle of the device's transmitting information to the server	1byte
RSSI	Indicating the signal receiving sensitivity between the device and the base station	1byte
Error Status	Storing the error code concerning the error and malfunction of transmitted information	1byte

4.2 Information Receiving Method

In the real-time container tracking device, information is periodically transmitted to the server by using TCP/IP socket communication, and in the server, it identifies the ACK(ACKnowledge) by decoding the encrypted packet which came through SEED encryption. At that time, the length of the encrypted packet is 32byte, and if it is normally decoded, it is arranged into 19byte, and the detailed explanation of the received payload is as follows.

Table 2. Payload in Information Reception from the Server

Section	Definition	Length
Packet Length	Length of Payload received from the real-time location tracking device	1byte
Protocol ID	ID of the relevant protocol. The ID of this protocol is 0x01	1byte
ConTracer ID	Unique number identifying TSR tag	8byte
Setting Change Flag	Making sure whether the setting is changed or not by the server during the operation of the real-time location tracking device	1byte
Optional Data	The information for changing equipment issuance information, which contains server IP/Port, checksum, etc.	8byte

4.3 Method to Store the Interior Memory of the Real-time Container Tracking Device

After transmitting information or in case of not transmitting information normally, the real-time container tracking device stores the information in the internal memory by using a mobile communication network(GSM or WCDMA). At that time, if GPS information value is not valid, the values of date and time is stored after using RTC value in the device. At that time, the storage capacity is set up to use it for 10 days continually on the basis of transmitting information at 1 hour intervals, and min. 240 logs or approx. 2,000 logs can be stored.

EEPROM used for the real-time container tracking device has a limited memory, and therefore, when the overflow of stored information such as more than 2,000 logs occurs, it does not store information anymore and inform the server of the memory overflow status through 'error identification field'. Until a user clears, it does not store information and get rid of it. As the real-time container tracking device has the information of protocol ID and tag ID that it transmits, in case of the information transmitted actually, only the payload(31byte) described in the following Tab. 3 is stored in the its memory.

Table 3. Map of the Transmitted Information Storage Memory of the ConTracer

Section	Length	Section	Length
Index	2 byte	Status of Container Door	1 byte
Date & Time	6 byte	Battery Residual Capacity	1 byte
Location Data	11 byte	Device On/Off Status	1 byte
Temp	2 byte	Transmission Cycle	1 byte
Humidity	2 byte	RSSI	1 byte
Impact	2 byte	Error Status	1 byte

5 CASE STUDY

5.1 Outline

As explained above, the container tracking system with the real-time container tracking device and the communication method between servers was applied to actual work-site operations. Container tracking system was applied to the combined district that marine transportation and land transportation(TSR: Trans Siberia Railway) from Korea to Moscow, Russia. Especially, TSR is the longest railway in the world and it connects the distance of 9,300km between Moscow and Vladivostok, and has harsh transportation environments that the temperature is 30 below zero in winter and so on.

Table 4. Application Environments of Real-Time Container Tracking System

Section	Definition	
Objects	LCL Freight. two 40ft containers	
Operational Period	Dec. 16, 2011 (Fri.) ~ Jan. 19, 2012 (Thu.), for 35 days	
Test Method (Selection of Routes and Equipments)	Yangsan ICD in Korea ~ Busan New Port in Korea ~ Vostochny Port in Russia ~ Moscow, Russia Among 100 devices, 2 devices were selected at random.	
Real-time container tracking device (ConTracer) ID	ID 0089	Information Transmission Cycle : 1 hour in the country, 10 hours in other countries (Used USIM: Vodafone)
	ID 0227	Information Transmission Cycle : 1 hour in the country, 1 hours in other countries (Used USIM: Megafon)

5.2 Analysis of Location Information Collected in ConTracer

In the result of operating the information transmitting/receiving method of the container tracking system(communication standards and data format), the location information of the container could be acquired as the below figure shown, and the information for global container transportation management which is the purpose of the container tracking system could be collected.

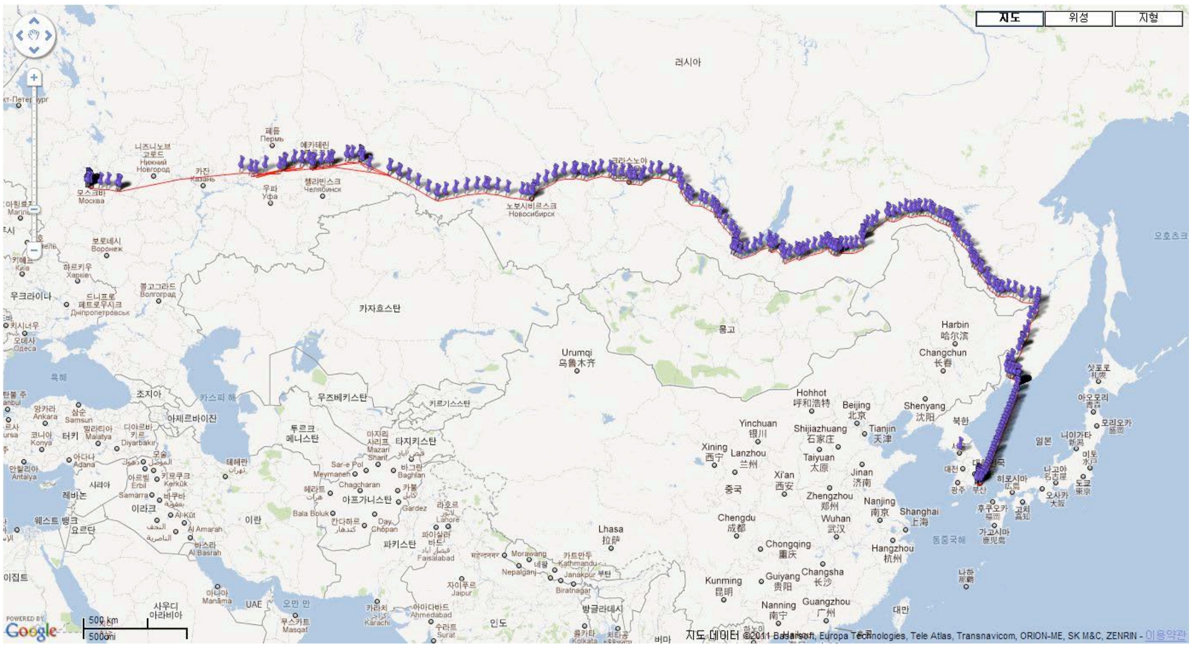


Figure 5. Location Information of Real-time container tracking device ID 0089

Especially, as shown in the result of location information collection, it was identified that the shadow zone occurred between Kazan, Russia and the surroundings of Moscow (approx. 1000km). In the shadow zone, the information could not be collected, but the transportation information can be collected by analysing the logs stored in the internal memory of the real-time container tracking device. In this area, it is assumed that due to the lack of communication infrastructure, real-time communication does not work, and it is necessary to seek for solution by expanding the communication infrastructure in the future or supporting it on business processes.

Figure 6. Location Information of Real-time container tracking device ID 0227

5.3 Analysis of Temperature and Humidity Information Collected in ConTracer

On the TSR transportation route from Korea to Russia, the temperatures collected in ConTracer are as shown in Fig. 7.

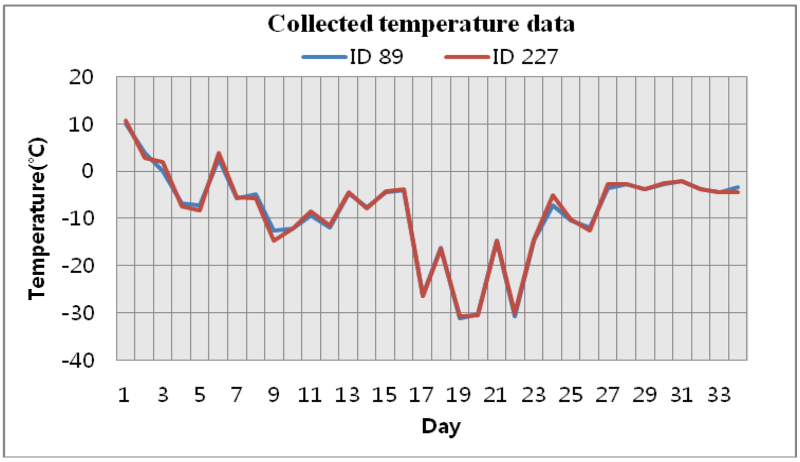


Figure 7. Analysis of temperature data

In Korea which is the starting point of the transportation, both ID 89 and ID 227 shows the temperatures above zero, but in transporting in Russia, the temperatures of ID 89 and ID 227 were measured up to min. -32 and the values of ID 89 and ID 227 are almost identical. Actually, when the difference of temperatures is severe, frost formation and damages occurs on the products(LCD and electronic appliances) packed in container, and in order to judge insurance coverage and responsibility, transport companies and consignors want to know when the cargo was damaged by the difference of temperatures. In addition, they want to have the information for electing future transportation routes to prevent the cargo from being damaged by the difference of temperatures by making use of the temperature information collected by routes.

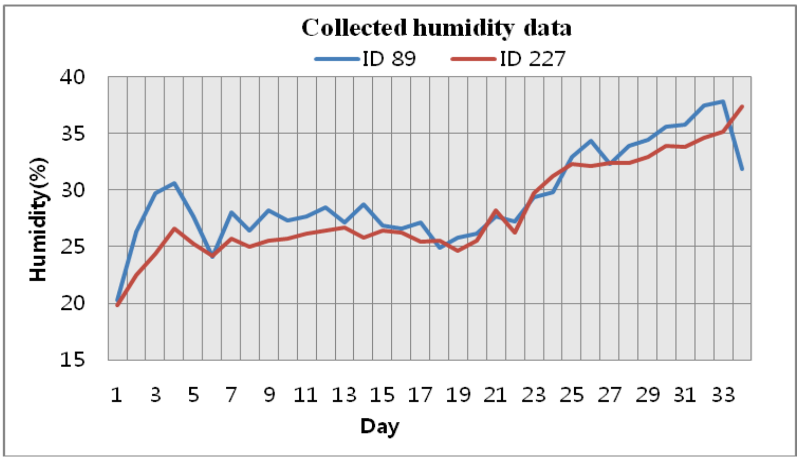


Figure 8. Analysis of temperature data

In case of humidity, both ID 89 and ID 227 show 40% which is general winter humidity. The reason why the internal humidity is important is that when damaged containers are used, snow or rainwater can be permeated into broken spaces and the cargoes can be wet. Therefore the value of product can be depreciated, so consignors and transport companies need the humidity information of container interior to judge the damages of their cargoes and prepare for future claims.

5.4 Analysis of Battery Residual Capacity of ConTracer

In the result of starting from Busan Port, Korea, trans-shipping into TSR in Vladivostok, attacked fully-charged ConTracer in the container transported to Moscow, and analysing the battery residual capacity of ConTracer, ID 89 and ID 227 showed different battery residual capacities.

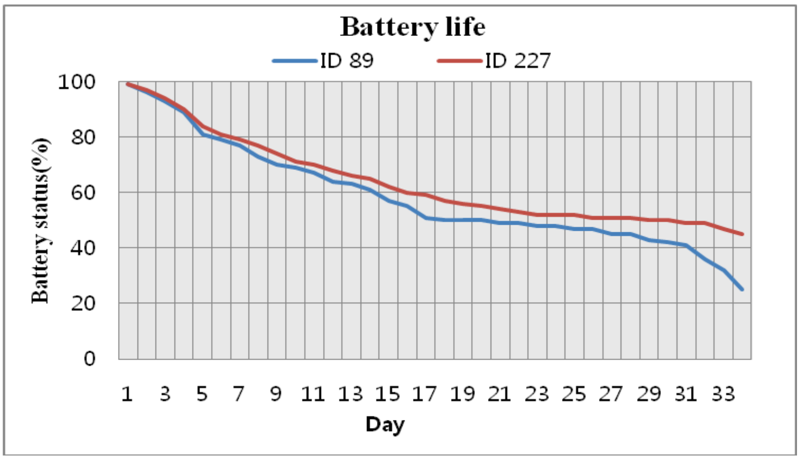


Figure 9. Analysis of temperature data

ID 89 and ID 227 had the same specification, and in ID, the USIM of Vodafone was used and in ID 227, the USIM of Megafon was used. As the above picture, the transportation distance of TSR is about 9,300km, and as the communication companies are different in each area of Russia, roaming among the companies can happen even in the same country.

As Fig. 9, in ID 227 using the USIM of Megafon(Russian communication company), the more battery capacity was measured than ID 89. The reason why battery residual capacities are relatively different is because, as ID 227 uses the communication company within Russia and its access of mobile communication is smoother than ID 89, the battery consumption was lower. As the communication company used by ID 89 was not a Russian company, it could be assumed that more access was tried than ID 228 and its battery consumption was higher.

Actually, in the result that applying the real-time container tracking device in the field, it is necessary to has maximum battery efficiency for tracking the containers that are transported in the world, and in order to use batteries efficiently, it seems more effective to use the communication company of the country where the container is transported.

6 CONCLUSION

This research investigated the composition plan of container tracking system, the real-time container tracking devices, and the communicating method of communication standards and data format with servers collecting information. By applying the proposed method to the actual container transportation routes, the information of location, temperature and humidity of the containers which are transported to the world could be collected in real time, and it is expected to propose the guideline to develop the container tracking system which is available in logistics service companies in the future. In addition, it is expected that the limitation of the previous technologies such as RFID, bar code, EDI and so on and the real-time global container tracking which could not be solved due to the limited logistics environments can be solved.

The case study of container management system proposed in this research has the limitation that it was applied only to the container transportation routes between Korea and Russia. In future study would apply the system to various transportation routes and show that in applying the proposed communication standards and data format to various M2M based logistics industries, it can be generally used.

ACKNOWLEDGEMENTS

This work (Grants No. C0038534) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

REFERENCES

Park, J.K. (2011), Future SCM Strategy, *Books21 Press*, Seoul, Korea

Anuparp, B., Park, S.K., Shin, S.H. (2011), An Approach of False Data Identification Protocol for Minimum Communication Cost in Wireless Sensor Network, *The Journal of Korea Society of Computer and Information*, **16(10)**, 121-129

Bae, W.S., Lee, J.Y., Kim, S.C. (2011), A New Low-Cost Mutual Authentication Protocol in RFID based on Hash Function, *The Journal of Korean association of computer education,* **14(1)**, 175-185

Choi, Y. B., Kim, Y.H., Kim, J.J., (2011), A Cost-Efficient Energy Supply Sources Deployment Scheme in Wireless Sensor Networks, *The Journal of the Korea Information and Communications Society*, **36(6)**, 738-743

Kim J.T. (2011), Analyses of Light-weight Protocol for Tag Security in RFID System. *Journal of the Korea Institute of Information and Communication Engineering*, **15(11)**, 2399-2403

Lee C.S, Kim, D.H., Kim, J.D. (2010), Design and Implementation of Large Tag Data Transmission Protocol for 2.4GHz Multi-Channel Active RFID System. *Journal of the Korean Institute of Information Scientists and Engineers,* **37(2)**, pp.167-253

Lee, C.Y., Hong, H.K. (2010), A Study of the Message Protocols Technologies in M2M Platforms, *Journal of Korea Information and Communications Society,* **35(1)**, 53-61

Min, Y.H (2011), Optimal Data Compression Algorithm Selection for Efficient Data Store and Transfer, *Journal of Korea Information and Science*, **38(2)**, 107-115

Bae, W.S., Lee, S.Y., Kim, Y.J. (2009), An Efficient Authentication Protocol for e-Seal Security, *Autumn Conference on the Korea Academia-Industrial Cooperation Society*, 591-593, Korea,

Julia, C. (2011), Smart Container Management: Creating Value from Real-Time Container Security Device Data, *IEEE International Conference on Technologies for Homeland Security(HST)*, 457-465

Kim S.J. (2008), Smart Container Management: Intelligent Networked Containers for Enhancing Global Supply Chain Security and Enabling New Commercial Value, In: *3rd International Conference on COMSWARE*, 662-669, Bangalore, India

Yingli W. (2008), The application of real time tracking technologies in freight transport, *Third International IEEE Conference on Signal-Image technologies and Internet-Based System*, 298-304, Bali, Indonesia ,

PROPORTIONAL EGALITARIAN CORE SOLUTION FOR PROFIT/COST ALLOCATION GAMES WITH AN APPLICATION TO COLLABORATIVE TRANSPORTATION PLANNING

Bo Dai[1] and Haoxun Chen[1]

[1]*Laboratoire d'Optimisation des Systèmes Industriels, Institut Charles Delaunay and UMR, CNRS STMR 6279, Université de Technologie de Troyes, 12 rue Marie Curie, BP 2060, 10010, Troyes Cedex, France, Dr.boDai@gmail.com, haoxun.chen@utt.fr*

ABSTRACT

For a profit/cost allocation game, one key issue is to find a fair allocation of profit/cost among its players. In this paper, a Proportional Egalitarian Core solution is proposed, which aims to find an allocation that takes account of the contribution of each player in the core of the game. This solution concept generalizes the Egalitarian-core solution. A row generation method is proposed to compute the new solution. As a case study, the solution concept is applied to a carrier collaboration problem with numerical results on randomly generated instances.

Keywords: Profit/cost allocation games, game theory, Proportional Egalitarian Core solution, collaborative transportation planning

1 INTRODUCTION

In profit/cost allocation games, multiple players cooperate together to finish a set of tasks using their resources. Usually, the total profit (resp. total cost) of the players obtained with the cooperation is higher (resp. lower) than their total profit (resp. total cost) without cooperation. One problem arisen in such a game is to fairly allocate the total profit/cost generated from the cooperation among the players. A number of papers have studied the problem. Some of them study profit allocation (Arin and Inarra, 2001; Arin et al., 2008; Dai and Chen, 2011b, 2012), and the others consider cost allocation (Tijs and Driessen, 1986; Caprara and Letchford, 2010; Drechsel and Kimms, 2010; Frisk et al., 2010). These two problems are related, since the total profit plus the total cost is the total revenue which is given for the players if the set of tasks to fulfill is given. Without loss of generality, we only discuss a profit allocation game in this paper, and call it as PAG for short. It is easy to modify the model and the method proposed in this paper to deal with cost allocation. Game theory is a natural tool to study the PAG. Most proposed allocation methods are based on concepts in cooperative game theory, which are briefly reviewed in the following.

In a given n-players game with players indexed by 1 to n, let $N = \{1, \ldots, n\}$ represent the grand coalition containing all players with a value (profit) $r(N)$, and S represent any subcoalition of N containing some players with a value (profit) $r(S)$. The problem in study is to fairly allocate the total value (profit) of the grand coalition, i.e., $r(N)$, among all players. An allocation of the profit is also called a solution. A fair allocation must satisfy some properties, such as Pareto optimal and individually rational. An allocation is called Pareto optimal if there is no other allocation in which each player can get a payoff (profit) larger than that in the considered allocation. An allocation is individually rational if each player can get from the coalition a payoff (profit) at least the same as it can get by acting individually. Note that the "profit" mentioned in this paper may mean the payoff of each player in a cooperative game, so the two terms "profit" and "payoff" are exchangeable hereafter.

For a cooperative game, an imputation (Kalai, 2008) is a feasible allocation (payoff vector) with Pareto optimal and individually rational properties. The core of the game (Gillies, 1959) is a set of imputations under which no coalition has a value greater than the sum of its members' payoffs. If an imputation is in the core, no player has an incentive to leave the grand coalition in order to receive a larger payoff. The core allocations are very important since only such allocations are stable. However, the core of a game may be contain more than one allocation and may be empty. In case that the core contains multiple imputations (allocations), a method is needed to choose one imputation (allocation) from the core.

Kalai-Smorodinsky solution (KS for short, Kalai, 1977) is another well-known concept for cooperative games with Pareto optimal property. For a game with two players, assume that player *i* (*i* = 1, 2) receives a payoff d_i without cooperation of the other player and receives a maximum payoff I_i with the cooperation of the other player, the KS solution (u_1, u_2) is the allocation satisfying the condition $(u_1 - d_1)/(I_1 - d_1) = (u_2 - d_2)/(I_2 - d_2)$, where u_i denotes the actual payoff of player *i*, *i* = 1, 2. Kalai (1983) generalized this solution concept to *n*-player game with the number of players fixed to *n*. In this case, the KS solution $(u_1, u_2, ..., u_n)$ satisfies $(u_k - d_k)/(I_k - d_k) = (u_m - d_m)/(I_m - d_m)$ for any $k, m = 1, ..., n,\ k \neq m$. KS solution is the unique imputation under which the payoff increase of each player with the cooperation of all other players is proportional to the difference between its maximum possible payoff I_i and its disagreement payoff d_i, but this imputation may be not in the core.

Shapley value (Shapley, 1953) defines an imputation according to the average marginal contribution of each player to each possible coalition, which is evaluated by its potential to alter the value of each coalition through joining or leaving the coalition. Marginal contribution of a player $k \in S$ to coalition *S* is defined as $MC_k(S) = r(S) - r(S \setminus \{k\})$. Shapley value is based on a particular concept of fairness in distributing the total value of the grand coalition (Forgó et al., 1999). Assume that all players enter the grand coalition one by one in some order, then there are *n*! possible orders. If these orders have the same probability to be chosen, then the average marginal contribution of player *k* is defined as $\phi_k = \sum_{S \subset N} MC_k(S) \cdot (s-1)!(n-s)!/n!$, *n*=|*N*|, *s*=|*S*|. For any coalition *S* containing player *k*, the probability that player *k* is the next player to enter the coalition $S \setminus \{k\}$ is $(s-1)!(n-s)!/n!$. Shapley value cannot guarantee that the allocation is always in the core.

Egalitarian solution (Kalai, 1977) is the unique imputation that equally divides the total value of the grand coalition among all players. Equal surplus division solution (Van den Brink, 2007) first assigns to each player its individual profit and distributes the surplus profit equally among all players, the solution is also called the center of the imputation set value (Dragan et al., 1996). However, neither Egalitarian solution nor equal surplus division solution can be guaranteed to be in the core, i.e., the two allocations may be not stable.

Recently, Egalitarian-core solution concept was proposed, which seeks for an egalitarian allocation in the core (Arin and Inarra, 2001; Arin et al., 2008). An allocation *x* is an Egalitarian-core allocation if it is in the core and no other allocation *y* in the core is the result of an equalizing bilateral transfer with respect to *x*. For any two players *k* and *m* in a game, an efficient allocation *x*, and a real number $\alpha > 0$, (k, m, x, α) is an equalizing bilateral transfer (of size α from *k* to *m* with respect to *x*) if $x_k - \alpha \geq x_m + \alpha$ or $x_m - \alpha \geq x_k + \alpha$.

Nucleolus (Schmeidler, 1969; Lemaire, 1984) is a solution concept that minimizes the maximal *excess* over all possible subcoalitions of the game. For any given imputation *x*, the excess of *x* for a subcoalition *S* is defined as the gain that players in *S* can obtain if they withdraw from the grand coalition *N* under payoff *x* and instead take the payoff *r*(*S*). For the payoff vector $x = (x_1,\ldots,x_n)$ to be decided, mathematically, nucleolus is to minimize ε subject to $r(S)-\sum_{k\in S} x_k \le \varepsilon, \forall S \ne \phi, S \subset N$, $\sum_{k\in N} x_k = r(N)$ and $x_k \ge r(\{k\}), \forall k \in N$. For a game, there is one and only one nucleolus, and the nucleolus is in the core if the core of the game is not empty.

From the above discussion, we know multiple methods have been proposed for allocation of payoff (profit) among players with different interpretations of fairness. These methods can be classified into three types: methods that generate multiple imputations in the core, methods that generate a unique imputation in the core, and methods that generate a unique imputation which may be not in core. Naturally, the second type is the most preferable one. For this reason, in this study, we try to propose a profit allocation method of the second type as an alternative to Egalitarian-core solution concept and Nucleolus solution concept. The new solution seeks for a unique allocation in the core that considers the contribution of each player evaluated according to an agreed upon preference of all players. Such allocation can ensure the sustainability of the grand coalition and is fair with regard to the preference. The solution is defined by a model which minimizes the weighted sum of the square profit increases of all players subject to the constraints ensuring that the allocation is in the core. The common preference of all players is specified by the weight of each player in the solution concept. This solution concept is further extended to the case where the core of a game is empty. In this case, a γ-budget balanced allocation (Caprara and Letchford, 2010) is sought, where γ is a value between 0 and 1, which is maximized under the condition that the γ-approximate core is not empty. Numerical experiments on randomly instances of a carrier collaboration problem with pickup and delivery requests (CCPPD) demonstrate the potential of the solution concept to be applied in collaborative logistics.

The contribution of this paper includes three aspects: 1) Extend the Egalitarian-core solution to a more general Proportional Egalitarian Core solution, which provides us with more choices for considering the contribution of each player in a profit allocation game. 2) Propose a row generation method to compute the Proportional Egalitarian Core solution. Compared with the method that considers explicitly all core constraints whose number is exponential ($2^n - 1$), our method is able to find the solution with much less computation time. 3) Apply the proposed model and method to a practical cooperative game in collaborative transportation planning.

The rest of this paper is organized as follows. Section 2 proposes the Proportional Egalitarian Core concept and its specializations for general profit/cost allocation games. Section 3 a presents a row generation method to compute an optimal allocation of the concept. Section 4 applies the proposed concept to a type of collaborative transportation planning problem, and some numeric experiments are also conducted. Section 5 concludes this paper with remarks for future work.

2 PROPORTIONAL EGALITARIAN CORE SOLUTION CONCEPT

When designing a mechanism for a profit allocation game, two issues must be considered. One is the stability of the grand coalition. The other is the contribution of each player to the grand coalition. The first issue has been considered by the core concept. The second issue, which is related to fairness, is not easy to be addressed because different players may have different evaluations about the contribution of each player in the game. In practice, if multiple enterprises intend to collaborate in a project, before the collaboration, they often sign a contract of collaboration which defines the percentage share of the total benefit or loss of the project by each enterprise. Motivated by this, we propose a *Proportional Egalitarian Core Solution Concept* (*PEC*), which not only ensures that the allocation is stable, but also ensures that each player obtains a profit proportional to its contribution. To make an allocation stable, the core conditions (Gillies, 1959) are considered in *PEC*. To reflect the fairness of an allocation, we consider the contribution weight of each player in the grand coalition. This weight can be set according to a common preference of all players and is specified in their collaboration contract.

Consider a cooperative game with n-players, each player i is associated with a positive weight w_i, which measures the contribution of the player to the grand coalition of the game, $i = 1, \ldots, n$, $w = (w_1, \ldots, w_n)$. Suppose that the core of the game is not empty. For any two core allocations $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ of the game, it is obvious that $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$. We say y can be obtained from x through a bilateral transfer from player k to player m ($k, m \in \{1, \ldots, n\}$, $k \neq m$), if there is a positive real number α such that $y_k = x_k - \alpha$, $y_m = x_m + \alpha$ and $x_i = y_i$ for any $i \neq k, m$. This transfer is called a proportional equalizing bilateral transfer with respect to x and w if $y_k/w_k \geq y_m/w_m$, i.e., $(x_k - \alpha)/w_k \geq (x_m + \alpha)/w_m$. If y is obtained from x through a proportional equalizing bilateral transfer, then $\sum_{i=1}^{n} x_i^2/w_i > \sum_{i=1}^{n} y_i^2/w_i$. This result can be proven as follows.

Suppose that for two players k and m, $y_k = x_k - \alpha$, $y_m = x_m + \alpha$, and $(x_k - \alpha)/w_k \geq (x_m + \alpha)/w_m$, $\alpha > 0$, we have:

$$
\begin{aligned}
& y_k^2/w_k + y_m^2/w_m = (x_k - \alpha)^2/w_k + (x_m + \alpha)^2/w_m \\
& = x_k^2/w_k + x_k^2/w_m - 2\cdot\alpha\cdot x_k/w_k + \alpha^2/w_k + 2\cdot\alpha\cdot x_m/w_m + \alpha^2/w_m \\
& = x_k^2/w_k + x_k^2/w_m - 2\cdot\alpha\cdot(x_k - \alpha)/w_k - \alpha^2/w_k + 2\cdot\alpha\cdot x_m/w_m + \alpha^2/w_m \\
& \leq x_k^2/w_k + x_k^2/w_m - 2\cdot\alpha\cdot(x_m + \alpha)/w_m - \alpha^2/w_k + 2\cdot\alpha\cdot x_m/w_m + \alpha^2/w_m \\
& = x_k^2/w_k + x_k^2/w_m - \alpha^2/w_m - \alpha^2/w_k < x_k^2/w_k + x_k^2/w_m
\end{aligned} \tag{1}
$$

Since $x_i = y_i$ for any $i \neq k, m$, we have $\sum_{i=1}^{n} x_i^2/w_i > \sum_{i=1}^{n} y_i^2/w_i$.

The above result implies that if the core of a game is not empty, then any propositional equalizing bilateral transfer from one player to another with respect to a core allocation x and a positive weight vector w will decrease the weighted sum of the square profits of all players. In other words, there is a core allocation x^* such that no core allocation y is the result of a weighted equalizing bilateral transfer with respect to x^* and w. This core allocation x^* minimizes the weighted sum of the square profits of all players, i.e., minimizes $\sum_{i=1}^{n} x_i^2/w_i$.

Motivated by the above analysis, we propose a solution concept and its mathematical model for the considered cooperative game. Firstly, the notation used in the model is given as follows.

Indices

k, m: player index, $k, m = 1, \ldots, n$, where n is the number of players

Parameters

N: grand coalition, which has n players, $N = \{1, \ldots, n\}$

$r(\{k\})$: the profit of player k without any cooperation with the other players

$r(N)$: the total profit of the grand coalition with the cooperation of all players

$r(S)$: the total profit of the coalition S, $S \subseteq N$

w_k: the weight of player k

Here, w_k reflects the contribution of player k to the grand coalition in the cooperative game. Without loss of generality, we assume $w_k > 0$ for any player k, because if there is a player with no contribution to the grand coalition, i.e., w_k = 0, it can be excluded from the profit allocation of the coalition. With the normalization, we can also assume $\sum_{k=1}^{n} w_k = 1$.

Variables

x_k : the profit allocated to player k

x_k^{PI} : the profit increase of player k through cooperation, $x_k^{PI} = x_k - x_k^{'}$

With the above notation, the *Proportional Egalitarian Core solution* (*PEC*) with respect to a given weight vector $w = (w_1, \ldots, w_n)$ of all players is defined as the optimal solution of the following quadratic programming model *PEC*:

Model *PEC*:

$$Z^{PEC} = Min \sum_{k=1}^{n} x_k^2 / w_k \tag{2}$$

Subject to:

$$\sum_{k \in S} x_k \geq r(S), \forall S \subset N, \tag{3}$$

$$\sum_{k=1}^{n} x_k = r(N) \tag{4}$$

$$x_k \in {}^\circ, x_k \geq 0, k = 1, \ldots, n, \tag{5}$$

The objective function of the model is the weighted sum of the square profits of all players. Constraints (3), which ensure the individual rationality of each player and the stability of the grand coalition, are called non-blocking constraints. Constraint (4), which implies the profit allocation is efficient (budget-balanced), is called distribution constraint. Constraints (3) and (4) together keep that the allocation is in the core.

In the above model, we assume the total profit of the grand coalition is allocated (distributed) among its players. This is one way of profit allocation. Another way is to allocate (distribute) among all players only the profit increase of their grand coalition due to the cooperation among them. Here, the profit increase of a coalition is defined as its profit gained through the cooperation of the players in the coalition minus the sum of the profit that each player can gain without any cooperation with other players. For defining the profit allocation in the second way, the objective function (2) of model *PEC* must be changed to (6), which is to minimize the weighted sum of the square profit increases of all players.

$$Z^{PEC} = Min \sum_{k=1}^{n} \left(x_k - r(\{k\}) \right)^2 / w_k \tag{6}$$

Obviously, the Proportional Egalitarian Core solution concept generalizes the *Egalitarian-core solution concept* proposed by Arin et al. (2001, 2008). Given a core allocation x, x is egalitarian if no other core allocation y is the result of an equalizing bilateral transfer with respect to x. The Egalitarian-core solution can be defined as the optimal solution of the following quadratic programming model *EC*:

Model *EC*:

$$Z^{EC} = Min \sum_{k=1}^{n} x_k^2 \tag{7}$$

Subject to constraints (3), (4), (5)

For model *PEC*, if we set $w_k = 1/n$ for $k = 1, \ldots, n$, it is transformed into model *EC*. Therefore, *EC* is a special case of *PEC*, i.e., Egalitarian-core solution is a special case of Proportional Egalitarian Core solution.

Furthermore, let $r^{\max}(\{k\})$ be the maximum profit that player *i* can obtain from the grand coalition in the cooperative game. If there is a proportional allocation (KS allocation) *x* such that

$$\left(x_k - r(\{k\})\right) \Big/ \left(r^{\max}(\{k\}) - r(\{k\})\right) = \left(x_m - r(\{m\})\right) \Big/ \left(r^{\max}(\{m\}) - r(\{m\})\right), \tag{8}$$

for any $k, m = 1, ..., n,\ k \neq m$,

and *x* is in the core, then *x* is the proportional egalitarian core solution with respect to the weight vector $w = (w_1, \ldots, w_n)$, where $w_k = r^{\max}(\{k\}) - r(\{k\})$, $k = 1, \ldots, n$. This is because if we take $w_k = r^{\max}(\{k\}) - r(\{k\})$ in the objective function (6) of model *PEC*, we can easily prove that any allocation satisfying the equation (8) is also an optimal solution of the model, it is thus a proportional egalitarian core solution with respect to *w*.

Our solution concept *PEC* provides a method to choose an imputation (allocation) from the core. Compared with Shapley value, the imputation defined by *PEC* is always in the core. Compared with nucleolus and the Egalitarian Core solution, *PEC* provides more freedom for the choice of an imputation from the core by allowing the definition of the (contribution) weight of each player by the grand coalition; furthermore, the determination of the weights requires a consensus of all players about their contributions in the game.

3 COMPUTATION OF PROPORTIONAL EGALITARIAN CORE SOLUTION

As the number of non-blocking constraints (3) in model *PEC* is exponential ($2^n - 1$), it is computationally inefficient to explicitly consider all constraints in solving the model when *n* is large. With this viewpoint, we propose a row generation method to compute the PEC allocation, the solution of the model. Row generation method has been applied to computing other core allocations defined by linear programs in cooperative games (Drechsel and Kimms, 2010). Our study extends the application of this method to the computation of a core allocation defined by a convex quadratic program.

3.1 Row generation method

It is an iterative procedure which solves two problems alternately. One is called master problem and the other is called subproblem.

The master problem is a restricted (relaxed) version of model *PEC*, which only considers part of core constraints (3). Let Ω be a set of coalitions (a set of subsets of *N*) considered, formally we can formulate master problem *MP* as the following quadratic program.

Model *MP*:

$$Z^{MP} = Min \sum_{k=1}^{n} x_k^2 \big/ w_k \tag{9}$$

Subject to:

$$\sum_{k \in S} x_k \geq r(S), \forall S \in \Omega, \tag{10}$$

$$\sum_{k=1}^{n} x_k = r(N) \tag{11}$$

$$x_k \in {}^{\circ}\ , x_k \geq 0, k = 1, ..., n, \tag{12}$$

The above model is the same as model *PEC* except that constraints (3) are replaced by constraints (10).

In the case of allocating the profit increase of the grand coalition among all players, we can revise the objective function (9) to (13) for the master problem.

Model *MP*:

$$Z^{MP} = Min \sum_{k=1}^{n} \left(x_k - r(\{k\})\right)^2 \Big/ w_k \tag{13}$$

The subproblem is to find a new coalition $S' \notin \Omega$ with $S' \neq \emptyset$ such that the allocation x_k, $k = 1, 2, \ldots, n$ given by the current master problem described by equations (9) to (12) violates its corresponding core constraint $\sum_{k \in S'} x_k \geq r(S')$ as much as possible. Such a coalition can be identified by taking $\sum_{k \in S'} x_k - r(S')$ as the objective function of *MP*. If the function objective has a negative optimal value, the coalition is found; otherwise, no such coalition exists. Let us define:

Parameters

$\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots \bar{x}_n)$: the optimal solution of the master problem

Variables

z_k: binary variable to indicate whether player k is in coalition S', $z_k = 1$ if $k \in S'$ and 0 otherwise.

The subproblem (SP) can be formulated as the following integer program.

Model *SP*:

$$Z^{SP} = Min \left\{ \sum_{k=1}^{n} \bar{x}_k \cdot z_k - r(S') \right\} \tag{14}$$

Subject to:

$$z_k \in \{0,1\}, k = 1, \ldots, n, \tag{15}$$

where $r(S')$ is the total profit (or the total profit increase) of coalition S' obtained with the cooperation among its players.

Note that, when applying the row generation method to the profit allocation of a particular cooperative game, we need to formulate $r(S')$ explicitly with additional problem-specific decision variables and constraints depending on the game. This will be explained through an application example on collaborative logistics in the next section.

With the above formulations of master problem and subproblem, the iterative row generation procedure can be summarized as follows:

1. Define an initial coalition set Ω, e.g., Ω = {{1}, …, {*N*}}.
2. Solve *MP* optimally and obtain a provisional allocation *x*.
3. If *MP* is infeasible then stop. The game has an empty core.
4. Otherwise, solve *SP* optimally to find a nonempty coalition $S' \notin \Omega$.
5. If such coalition S' exists, i.e., the optimal objective value of *SP* is negative, then update $\Omega = \Omega \cup \{S'\}$ and return to step 2.
6. Otherwise, the current allocation *x* is the Proportional Egalitarian Core allocation.

3.2 Case with empty core

If the core of the profit allocation game is empty, model *PEC* and model *MP* have no solution. In this case, instead of finding a core allocation, we seek for a γ-budget balanced allocation (Caprara and Letchford, 2010), where 0 < γ ≤ 1 is a positive number. The maximum value of γ can be obtained by solving the following model *BBA*.

Model *BBA*:

$$Z^{BBA} = Max\, \gamma \tag{16}$$

Subject to constraints (4), (5) and:

$$\sum_{k=1}^{n} x_k \geq \gamma \cdot r(N) \tag{17}$$

$$\gamma \in \mathbb{R}, \gamma \in (0,1] \tag{18}$$

Let γ* be the maximum value. The γ*-budget balanced allocations always exist but may be not unique. We can then define the unique γ*-budget balanced allocation which is proportional egalitarian in a similar way as the *PEC* allocation. Given a weight vector *w* of all players, a γ*-budget balanced allocation *x* is proportional egalitarian if no other γ*-budget balanced allocation *y* is the result of a proportional equalizing bilateral transfer with respect to *x* and *w*. The γ*-budget balanced proportional egalitarian allocation can be obtained by the following two steps:

1. Solve model *BBA* by using the row generation method to get the maximum value γ*.
2. Replace constraints (4) in model *PEC* by constraints (17) with γ = γ* and solve the model by using the row generation method. The solution of the modified model is the γ*-budget balanced proportional egalitarian allocation.

4 APPLICATION OF PROPORTIONAL EGALITARIAN CORE SOLUTION CONCEPT

In this section, the *Proportional Egalitarian Core solution concept* is applied to a profit allocation game appeared in carrier collaboration, where multiple carriers form a collaborative alliance by sharing their transportations requests and vehicle capacities. The objective of the collaboration is to increase the profit of each carrier by eliminating empty back hauls and raising vehicle utilization rates.

4.1 Carrier collaboration problem

We consider a carrier collaboration problem in pickup and delivery service (CCPPD) (Dai and Chen, 2011a, 2011b, 2011c, 2011d, 2012), where multiple carriers operating in a transportation network form a collaborative alliance to share their transportation requests and vehicle capacities. It is assumed that initially each carrier has acquired certain transportation requests from its customers (shippers) and each request is specified by a pickup location, a delivery location, a pickup/delivery quantity, and the time windows associated with the pickup and delivery operations of the request. With the collaboration, each carrier makes its transportation requests available to all carriers in the alliance and a collaborative transportation planning is made, which reallocates all the requests among the carriers so as to maximize the total profit of the alliance. After the implementation of the collaborative transportation plan by the carriers, the total profit will be allocated among them in a fair way taking account of the contribution of each carrier in the realization of the plan. The decision problem for the collaborative transportation planning can be formulated as a mixed integer program (MIP) (Dai and Chen, 2011a). The profit or cost allocation among carriers in carrier collaboration has been discussed by Krajewska and Kopfer (2006), Krajewska et al. (2008), Frisk et al. (2010), Audy et al. (2011), and Houghtalen (2011).

4.2 Proportional egalitarian core allocation for the carrier collaboration

To define a proportional egalitarian core allocation for the carrier collaboration problem, we first have to define a weight vector for all the carriers involved in the collaboration. The weight of each carrier measures its contribution to the carrier alliance (the grand coalition). Since each carrier can contribute to the alliance through offering requests to other carriers and serving requests of other carriers, we define the weight of each carrier *k* as

$$w_k = \left(\theta_1 \cdot c_k + \theta_2 \cdot R_k\right) \Big/ \left(\theta_1 \cdot \sum_{k=1}^{n} c_k + \theta_2 \cdot \sum_{k=1}^{n} R_k\right), k = 1, \ldots, n, \tag{19}$$

where θ_1 and θ_2 are two positive coefficients (weights) set before profit allocation (Dai and Chen, 2012). The reader is referred to the paper for setting c_k, R_k, θ_1 and θ_2.

4.3 Computation of the PEC allocation using the row generation method

The PEC allocation defined in section 4.2 for carrier collaboration can be computed by using the row generation method proposed in section 3.1. At first, the master problem is formulated as model *MP* presented in section 3.1. Then the subproblem *SP* is redefined based on collaborative transportation planning model *CTP* for the case where all requests are served by the carrier alliance (Dai and Chen, 2011a), where some new notations are introduced.

Parameters

RO_k : the set of requests offered by carrier k

Variables

z_k : 1 if carrier k is in coalition S' and 0 otherwise

Model *SP*:

$$Z^{SP} = Min\left\{\sum_{k=1}^{n}\overline{x}_k \cdot z_k - obj\right\} \tag{20}$$

Subject to:

$$obj = \sum_{k=1}^{n}\sum_{l=1}^{L} p_l \cdot y_{lk} - \sum_{k=1}^{n}\sum_{i=1}^{H}\sum_{j=1, j\neq i}^{H} c_{ij}^k \cdot x_{ij}^k \tag{21}$$

$$\sum_{m=1}^{n} y_{lm} \le z_k, k = 1,...,n, l \in RO_k, \tag{22}$$

$$\sum_{j=1, j\neq o_k}^{H} x_{o_k j}^k \le W_k \cdot z_k, k = 1,...,n, \tag{23}$$

$$z_k \in \{0,1\}, k = 1,...,n, \tag{24}$$

and all constraints except for constraints (22) in model *CTP*

Then the PEC allocation for all carriers can be obtained by iteratively solving *MP* and *SP* following the steps of the row generation method presented in section 3.1.

4.4 Numeric experiments

To our best knowledge, there are no benchmark instances for the carrier collaboration problem we study. For this reason, we use 20 randomly generated instances to demonstrate our proposed PEC solution concept and to evaluate the performance of the row generation method. The first 10 instances are generated on a transportation network with 5 carriers and 35 nodes, where 5 nodes are the depot nodes of 5 carriers (one depot node for each carrier) and the other 30 nodes are customer nodes. The second 10 instances are generated on a transportation network with 5 carriers and 55 nodes (5 depot nodes and 50 customer nodes). All these instances are generated in the same way as in Dai and Chen (2011a, 2012).

All mixed integer programming models (*MP* and *SP*) involved in the computation of PEC allocation for each instance are solved by using Cplex 12 on a workstation 7550-XEON with 2GHz processor and 2Go RAM, where multiple processes may be activated and run by multiple users simultaneously. For the instances, all requests must be served. The results are given in Table 1 and Table 2, where *RGM* denotes row generation method. The row entitled 'Computation time without RGM (s)' gives the computation time for finding the PEC allocation based on the computation of the profit of each possible subcoalition.

Table 1. Results of ten random instances with 5 carriers, 15 requests and 35 nodes

Instance	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Pre-collaboration profit of each carrier	46.8 56.7 5.5 0 0	9.9 19 55 6.1 26.4	0.1 11.7 0 24.9 4.7	0 9 15.7 0 0	30.8 2.6 0 10.1 7.9	56.5 20.7 5.1 5.1 0	13.4 33.5 0 0.2 7.3	54.2 0 0 0 40.8	69.6 0 8.7 10.2 0	5.2 0 43 82 24
Pre-collaboration total profit	109	116.4	41.4	24.7	51.4	87.4	54.4	95	88.5	154.2
Post-collaboration total profit	315.7	420.9	340.4	259.2	266.3	274.4	258.8	643.7	328.7	431.8
Contribution weight of each carrier	0.166157 0.136229 0.290947 0.302521 0.104146	0.071428 0.196723 0.498972 0.080235 0.152642	0.257653 0.430794 0.089962 0.121212 0.100379	0.260164 0.133958 0.285776 0.116167 0.203936	0.575663 0.069973 0.089184 0.145161 0.120019	0.495253 0.066703 0.22004 0.10141 0.116594	0.108184 0.107623 0.023941 0.555226 0.205026	0.482535 0.105762 0.093922 0.054854 0.262927	0.210162 0.167172 0.098152 0.476016 0.048498	0.161913 0.289732 0.124326 0.244846 0.179183
Post-collaboration profit increase of each carrier by PEC	31.2792 36.101 54.771 56.9498 27.599	42.8437 43.1258 109.385 17.5892 91.556	46.7 132.4 34.6216 46.648 38.6304	74.4605 38.3395 51.3332 20.8668 49.5	83.9 25.3146 32.265 30 43.4204	59.5 20.0783 41.8 30.5255 35.0962	21.4113 21.3004 5.4165 109.888 46.3838	114.2 88.8048 78.864 46.0592 220.772	103.039 81.961 30.4 22.3 2.5	89.656 23.8 68.844 45.9 49.4
Number of iterations of RGM	7	7	8	8	8	8	7	7	11	9
Computation time of RGM (s)	502	1608	4202	1592	947	671	755	511	682	142
Computation time without RGM (s)	2061	5318	3968	3734	2938	1389	2246	2498	897	940

Table 2. Results of ten random instances with 5 carriers, 25 requests and 55 nodes

Instance	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Pre-collaboration profit of each carrier	129.9 0 20.4 34.5 5	195.8 173.2 49.8 0 0	9.9 48.6 86.7 0 34.2	66.1 0 72.6 233.8 25.2	124.1 5.9 57.5 14.7 191.2	23.8 30.7 109.4 0 0.1	0.2 59 36.2 28.8 57.2	35.5 92 286.1 0 0	0 0 17.6 121.6 91.5	23.7 101.2 0 0 168.4
Pre-collaboration total profit	189.8	418.8	179.4	397.7	393.6	164	181.4	413.6	230.7	293.3
Post-collaboration total profit	934	1128.8	894.9	1080.9	854.1	893.3	982.4	1009.5	1031.8	800.6
Contribution weight of each carrier	0.108749 0.140691 0.462051 0.160634 0.127875	0.122061 0.166004 0.094756 0.493221 0.123958	0.059772 0.178864 0.358631 0.069091 0.333642	0.118657 0.497005 0.119076 0.179518 0.085744	0.311367 0.074594 0.302767 0.134458 0.176814	0.390466 0.132114 0.13045 0.237471 0.109499	0.099045 0.170591 0.480818 0.138474 0.111072	0.09848 0.2875 0.288283 0.104855 0.220882	0.043687 0.117139 0.488769 0.249109 0.101296	0.12048 0.134167 0.138104 0.097584 0.509665
Post-collaboration profit increase of each carrier by PEC	90.161 116.644 298.2 133.177 106.018	172.161 234.139 69.8 149.9 84	82.883 248.017 187.3 61.8 135.5	103.75 243.4 104.116 156.964 74.97	167.064 40.0235 86.6 72.1435 94.869	256.751 112.349 110.934 156.149 93.117	102.215 176.051 265.2 142.907 114.627	59.074 169.559 172.928 61.841 132.498	44.752 119.996 277.4 255.185 103.767	90.888 101.213 104.183 73.616 137.4
Number of iterations of RGM	7	10	9	7	7	7	7	7	7	7
Computation time of RGM (s)	2131	2433	108289	18134	70933	52754	217280	2198	1798	43709
Computation time without RGM (s)						-				

From Table 1 and Table 2, we can see: 1) both the total profit of the carrier alliance and the profit of each carrier largely increase through collaboration. 2) a carrier with a larger contribution weight obtains a larger profit increase for 11 instances, whereas for the other 9 instances, the profit increases of some carriers are not strictly proportional to their contribution weights. This disproportionality is due to the core constraints of the PEC allocation. Our PEC solution concept interprets the fairness of an allocation not only by considering the contribution of each player defined by its weight but also the stability of the grand coalition guaranteed by the non-blocking constraints and the distribution constraint. 3) for the second set of 10 instances, we do not list the computation time without RGM since it is very time-consuming to find the PEC allocation by enumerating all possible sub-coalitions. Note that the row generation method can generate a solution for each tested instance, this implies that all profit allocations generated are in the core and thus stable.

Moreover, the computation time for finding a PEC allocation by the row generation method is much shorter than that based on the computation of the profit of each possible subcoalition. That is because the RGM method only runs fewer iterations, which implies that many subcoalitions are not considered when generating the PEC allocation. Because of computation time limitation, we did not test large instances, but we have a strong reason to believe that the row generation method is more efficient for large instances. Because the number of subcoalitions ($2^n - 1$) increases exponentially as the number of carriers (n) increases, however, the RGM, which is an implicit enumeration method, only enumerates few of the subcoalitions and the ratio of enumeration (the number of subcoalitions enumerated divided by the total number of subcoalitions) will decrease as n increases as most implicit enumeration methods.

5 CONCLUSION

A Proportional Egalitarian Core solution concept is proposed for profit/cost allocation games in this paper. It considers the contribution weight of each player participating in a game and is a generalization of the Egalitarian-core solution. To efficiently compute the PEC solution, a row generation method is developed. The potential of our solution concept to be applied in practice is demonstrated by a case study on a carrier collaboration problem and the performance of the row generation method is evaluated through randomly generated instances of the problem. In the future, we will apply our solution concept to other collaboration problems in logistics.

REFERENCES

Agarwal R., Ergun, O. (2008), Mechanism design for a multicommodity flow game in service network alliances, *Operations Research Letters*, Vol. **36**, No. 5, 520-524.

Arin, J., Inarra, E. (2001), Egalitarian solutions in the core, *International Journal of Game Theory*, Vol. **30**, No. 2, 187-193.

Arin, J., Kuipers, J., Vermeulen, D. (2008), An axiomatic approach to egalitarianism in TU-games, *International Journal of Game Theory*, Vol. **37**, No. 4, 565-580.

Audy, J.F., D'Amours, S., Rousseau, L.M. (2011), Cost allocation in the establishment of a collaborative transportation agreement: an application in the furniture industry, *Journal of the Operational Research Society*, Vol. **62**, No. 6, 960-970.

Caprara, A., Letchford, A.N. (2010), New techniques for the cost sharing in combinatorial optimization game, *Mathematical Programming Series B*, Vol. **124**, No. 1-2, 93-118.

Dai, B., Chen, H.X. (2012), Profit allocation mechanisms for carrier collaboration in pickup and delivery service, Computers & *Industrial Engineering*, Vol. **62**, No. 2, 633-643.

Dai, B., Chen, H.X. (2011a), A multi-agent and auction based approach and approach for carrier collaboration, *Logistics Research*, Vol. **3**, No. 2-3, pp.101-120.

Dai, B., Chen, H.X. (2011b), A Profit Allocation Mechanism for carriers' collaboration in less than truckload transportation, *International Conference on Industrial Engineering and System Management, IESM*, Metz, France.

Dai, B., Chen, H.X. (2011c), Price-setting based combinatorial auction approach for carriers' collaboration in less than truckload transportation. *The 3rd International Conference on Agents and Artificial Intelligence*, Rome, Italy.

Dai, B., Chen, H.X. (2011d), Mathematical model and solution approach for carriers' collaborative transportation planning in less than truckload transportation, *International Journal of Advanced Operations Management*, Vol. **4**, No. 1-2, pp.62-84.

D'Amours, S., Rönnqvist M. (2010), Issues in collaborative logistics, *Energy, Natural Resources and Environmental Economics*, Part 3, Springer Berlin Heidelberg.

Drechsel, J., Kimms A. (2010), Computing core allocations in cooperative games with an application to cooperative procurement, *International Journal of Production Economics*, Vol. **128**, No. 1, 310-321.

Forgó, F., Szép, J., Szidarovszky, F. (1999), Introduction to the theory of games: concepts, methods, applications, Springer Berlin Heidelberg.

Frisk, M., Göthe-Lundgrenc, M., Jörnstena**,** K., Rönnqvist, M. (2010), Cost allocation in collaborative forest transportation, *European Journal of Operational Research*, Vol. **205**, No. 2, 448-458.

Houghtalen, L., Ergun, Ö., Sokol, J. (2011), Designing allocation mechanisms for carrier alliances, *Transportation Science*, Vol. **45**, No. 4, 465-482.

Kalai, E. (1977), Proportional solutions to bargaining situations: Intertemporal utility comparisons, *Econometrica*, Vol. **45**, No. 7, 1623-1630.

Kalai, E. (1983), Solutions to the bargaining problem, *Discussion Papers 556*, Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston and Chicago, Illinois, USA.

Kalai, E. (2008), Games in coalitional form, *New Palgrave Dictionary of Economics*, second edition, available at (Accessed July 8, 2011): http://www.kellogg.northwestern.edu/research/math/papers/1449.pdf

Krajewska, M.A., Kopfer, H. (2006), Collaborating freight forwarding enterprises: request allocation and profit sharing, *OR Spectrum*, Vol. **28**, No. 3, 301-317.

Krajewska, M., Kopfer, H., Laporte, G., Ropke, S., Zaccour, G. (2008), Horizontal cooperation of freight carriers: request allocation and profit sharing, *Journal of the Operational Research Society*, Vol. **59**, No. 11, 1483-1491.

Lemaire, J. (1984), An application of game theory: cost allocation. *ASTIN Bulletin*, Vol. **14**, No. 1, 61-81.

Schmeidler, D. (1969), The nucleolus of a characteristic function game, *SIAM Journal on Applied Mathematics*, Vol.**17**, 1163-1170.

Shapley, L.S. (1953), A value for n-person games, *Annals of Mathematical Studies*, Contributions to the Theory of Games, Vol. **2**, No. 28, Princeton University Press, 307-317.

Tijs, S.H., Driessen T.S.H. (1986), Game theory and cost allocation problems, *Management Science*, Vol. **32**, No. 8, 1015-1028.

A FLEXIBLE OPTIMISATION MODEL FOR THE RAILWAY CREW SCHEDULING PROBLEM

Rosmalina Hanafi[1] and Erhan Kozan[2]

[1]*School of Engineering Systems, Science and Engineering Faculty, Queensland University of Technology*

[2]*School of Mathematical Sciences, Science and Engineering Faculty, Queensland University of Technology, Gardens Point Campus, 2 George Street Brisbane, Qld 4000 Australia, e.kozan@qut.edu.au*

ABSTRACT

Addressing the Crew Scheduling Problem (CSP) in transportation systems can be too complex to capture all details. The designed models usually ignore or simplify features which are difficult to formulate. This paper proposes an alternative formulation using a Mixed Integer Programming (MIP) approach to the problem. The optimisation model integrates the two phases of pairing generation and pairing optimisation by simultaneously sequencing trips into feasible duties and minimising total elapsed time of any duty. Crew scheduling constraints in which the crew have to return to their home depot at the end of the shift are included in the model. The flexibility of this model comes in the inclusion of the time interval of relief opportunities, allowing the crew to be relieved during a finite time interval. This will enhance the robustness of the schedule and provide a better representation of real world conditions.

Keywords: combinatorial optimisation, crew scheduling, mathematical programming.

1 INTRODUCTION

Crew scheduling problem (CSP) is a well known combinatorial optimisation in public transportation systems and is considered as a highly constrained problem. The main objective is scheduling the crews optimally while satisfying a number of constraints imposed by governmental regulations, union agreements, and company specific rules. Crew costs represent one of the largest components of operational expenditure. Developing a better, more feasible schedule is strategically important since small improvements to this cyclic schedule can lead to significant cost savings.

Railway crew scheduling is domain specific in which the constraints involved and dictate a problem might not be applicable to other situations. The developed models and algorithms are mainly designed for a particular condition and might not readily be adapted to another situation.

Due to its inherent computational complexity, a number of solutions have been proposed to handle the problem. The most frequent method of solving the CSP has been through modeling the CSP as a set covering problem (SCP) or set partitioning problem (SPP). In both the SCP and SPP formulations of the CSP, the decision variable is a binary integer variable that identifies whether or not a duty (pairing) is selected in the schedule. The constraints consist of a matrix of binary values that indicate if a pairing j covers a work duty i. Each row in this 0-1 matrix shows which pairings cover a single work duty. Each column corresponds to one possible pairing or the work for an individual crew member over the defined time horizon. The SCP formulation is more flexible, allowing for over coverage of work requirements while the SPP restricts the coverage of the task exactly once. The main difficulty with the SPP and SCP formulations of the CSP is that of determining all possible pairings for these models. For a CSP with a large number of trips, there can be an unmanageably huge amount of possible pairings. As a consequence, the problem becomes a time-consuming task of enumerating all the possible pairings. In the case of large scale combinatorial problems, this may not lead to feasible solutions within a finite time.

Caprara et al. (1997) formulated crew scheduling for the Italian railways as a set covering problem. The developed model followed the approach applied for airline crew scheduling. The variables are associated with the circuits and the edges of the graph which are solved by Lagrangean relaxation and a heuristic technique. A further study by Caprara et al. (2001) used constructive heuristics with relaxation techniques to solve the CSP. This research divided the CSP into three parts, i.e pairing generation, pairing optimisation, and roster optimisation. The problem is one of generating cost efficient rosters that cover all scheduled train trips. A depth-first branch and bound method is first employed to enumerate all feasible pairings for all depots. Heuristics are used to reduce the feasible pairing sets. The research experimentally shows that solution quality can be substantially improved if the pairing optimisation and roster optimisation phases of the process can be iterated through a feedback mechanism.

A set covering model was also applied by Freling et al. (2001) for scheduling train crews at Dutch Railways (NS). The side constraints correspond to the high level constraints dealing with sets of duties. Medium level constraints deal with the construction of paths during the column generation procedure, while low level constraints deal with the construction of the network. This research uses a column generation approach to solve the LP relaxation of the IP formulation and a branch-and-price heuristic to find integer solutions. New columns are generated implicitly using a dynamic programming algorithm. To be able to solve the large-scale train crews scheduling, the researchers devised their approach with several acceleration techniques to speed up the algorithm.

The case of scheduling train drivers on a railway subnetwork was presented by Alfieri et al. (2007). The scheduling problem involves the construction of feasible duties from a set of trips to be serviced by a number of train drivers. This research uses a set covering formulation based on an implicit column generation solution approach and focuses on minimising the number of duties and on maximizing the robustness of the obtained schedule for outside disruptions. A heuristic procedure is presented to find an initial feasible solution together with a heuristic branch-and-price algorithm based on a dynamic programming algorithm for the pricing-out of columns. This approach is tested on the timetable of the intercity train of NS Reizigers, the largest Dutch operator of passenger trains. Although the proposed approaches are considered acceptable, the findings suggest an improvement in the algorithm by studying further several issues such as how many columns are to be added in each iteration of the pricing algorithm for speeding up the convergence of the algorithm and how to find the best stop criterion for stopping the column generation in a certain node.

Bengtsson et al. (2007) also studied the crew pairing problem at the large European railway, Deutsche Bahn. A mathematical formulation is presented with the objective function being to minimise the cost of selected pairing and the cost of violating soft constraints. This research shows that a column generation approach to the pairing problem, which combines resource constraints, k-shortest path enumeration and label merging techniques, is able to heuristically solve large and highly complex railway pairing problems in a reasonable time. Given the size and complexity of the railway operation, the researchers indicate the necessity of tailored optimisation techniques.

More recently, Kwan (2010) presented a case study of an automatic optimising train crew scheduling system in UK, TrainTRACS. The optimisation technique is formulated mathematically as an integer linear program (ILP) based on set covering. Train crew scheduling is partitioned into segments which are permuted and recombined with breaks and other crew activities to form crew shifts.

Eventhough some work has been done on the CSP using a wide variety of solution techniques, the subject is still hard to solve. This paper presents an alternative formulation to the problem using MIP incorporating commonly encountered constraints of railway CSP. The main contribution of this work lies in the representation of the real condition of railway CSP in the model by including the time interval of relief opportunities which offers a flexibility for the crew in choosing a relief opportunity within this time frame.

2 PROBLEM DESCRIPTION

The railway crew scheduling involves a railway network where trains travel from a station to a subsequent station according to the published train schedule. There are depots in a railway network to which sets of crew members are allocated. Crew members are responsible for performing a set of activities to meet the schedule demands. The problem is to construct work schedules for crew members located in the depots such that they comply with the given rules and schedule demands.

A crew typically operates a train starting from the home depot and travelling from one station to the next. The crew rests for a specified time (relief time) at a specified location (relief point), and then operates another train back to the home depot. The railway crew scheduling problem in this context is to specify the sequence of trips to be performed by the crew. A shift is divided into a number of duties of different duration. The start and end of a duty are associated with the boundaries of the predefined time period. A duty consists of a sequence of train trips (*spell*) that begins at a crew home depot and ends either at the crew home depot or at a relief point, and can feasibly be serviced by a single crew. The set of crew home depots is a subset of the set of relief points. Two individual duties which are separated with a period of a meal break time will make one shift.

The passenger railway operator in this study is Queensland Rail (QR) which has several crew depots and offers regular service trips on specified lines according to a published train timetable. A line is characterized by a departure station and an arrival station and a number of intermediate stations. The trains will travel along this line from a station to a subsequent station. When the line is served by a single crew, the sequence of trips can be treated as individual trip. By treating this compound trip as a single trip, the size of the problem can be reduced significantly. An example of such a line is the 56.130 km line from Ferny Grove (FYG) to Beenleigh (BNH) with 38 intermediate stations. The trip from FYG to BNH in this study will be considered as a single trip and vice versa. This line mainly serves the train trip from FNY to BNH. However, there are also train trips from FNY to other terminal stations such as Cleveland (CVN), Corinda (CQD), and Bowen Hills (BHI). The same situation also occurs on the other lines in the QR network.

The train crew scheduling problem in this study has the following inputs. A set of home depots and a set of relief points, a set of train trips with fixed starting and ending times, and predetermined driving times between all pairs of stations.

A train crew shift contains a meal break (MB) which starts and ends within a specific period as determined in the union agreements. This meal break will begin after the completion of the third hour and finish before the completion of the sixth hour relative to the start of shift. For example, crews sign on at 0800 hours and sign off at 1600 hours. Then the earliest meal break will be at 1100 hours to 1130 hours and the latest meal break will be at 1329 hours to 1359 hours (see Figure 1). The time period between the earliest break and the latest break is the transition time between duties ($\zeta_{ii'j(j+1)k}$) and this is also known as relief opportunities. The relief opportunities are the time period in which a train crew is allowed to be relieved at any relief point. A relief opportunity can be chosen for a crew relief at any relief point within the two limits of this time interval.

Figure 1 illustrates two possible alternatives of crew relief. The first condition is when the meal break occurs for minimum time (30 mins) at the earliest time and the second condition is when the meal break occurs at the latest time.

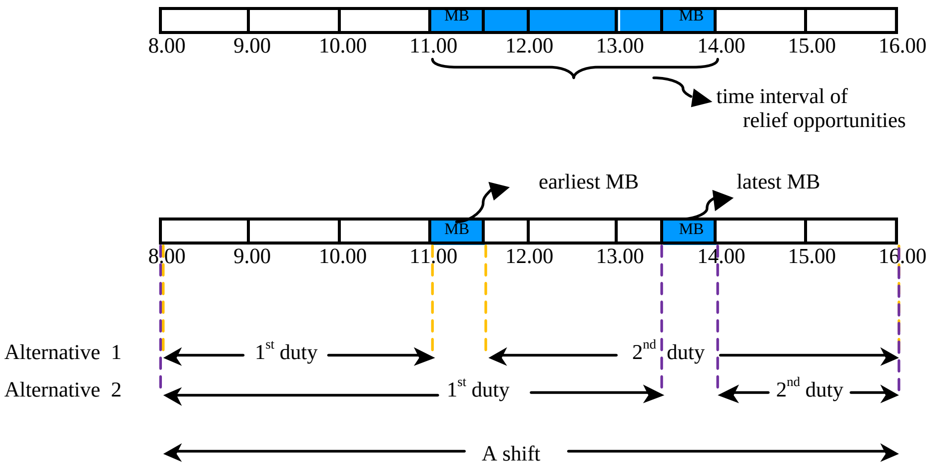


Figure 1. Two possible combination of duties in a shift

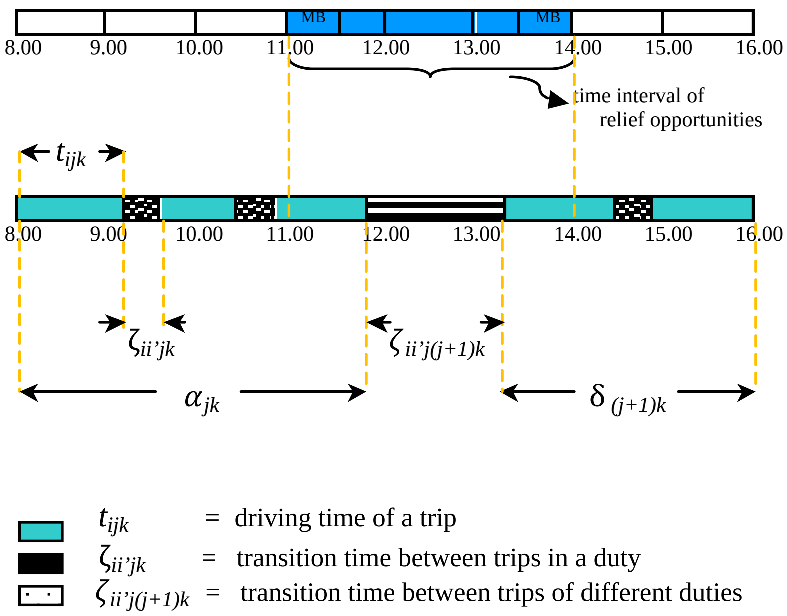


Figure 2. A sequence of trips in a shift

As can be seen in the Figure 1, if the crew relief is at the earliest time, the first duty is the period of time from 0800 to 1100 and the second duty is the period of time from 1130 to 1600. If the crew relief is at the latest time, then the first duty starts from 0800 to 1329 and the second duty starts from 1400 to 1600.

A sequence of trips in a shift is shown in Figure 2. As can be seen in this figure, the meal break divides the shift into two different durations of duty. The first duty (α_{jk}) is the period from start of a shift to the start of the meal break. Whereas the second duty ($\delta_{(j+1)k}$) is the period from the end of the meal break to the end of the shift. Every single horizontal blue bar in Figure 2 represents a trip (t_{ijk}). Transition time ($\zeta_{ii'jk}$) or *turn around time* in a duty is the time incurred between trips which includes the time required by the crew to move from one end to the other end of the train at platform and drives the train away in the opposite direction. Transition time ($\zeta_{ii'j(j+1)k}$) in a shift is the time period between trips of different duties which includes the meal break and crew relieving related activities. A duty in this model is a period of time a train crew drives continously without a meal break. There is a limitation on the maximum length of *continous driving* and *total continous driving* time (aggregate driving time) in a shift.

The approach to solve the railway CSP follows several stages. Firstly, the train segment is divided into trips for each depot. This is because the trip combinations (pairing) can be significant. A duty is defined as a sequence of trips which can be assigned to a crew. The optimisation model sequences the trips into feasible duties and minimises the total elapsed time of the duties (shifts). The best subset will then be selected from all the generated duties based on the minimum cost of duties. In the crew rostering phase, the duties selected in the previous phase will be sequenced to obtain the roster. A periodic duty assignment to each crew will then be defined in which all duties are covered for a certain number of consecutive days (a week).

Union agreements and company policies impose a complex set of restrictions in the formation of shifts. In particular, each shift should satisfy several constraints corresponding to work load regulations for crew. Such constraints are incorporated in the model and are given as follows.

- There are minimum and maximum duration of a shift which is determined from the starting time of the shift.
- The working time of the duty has to be within a specified limit.
- There is a maximum duration of continous driving time. A crew will be required to take a break when the continous driving time has reached a certain limit.
- There must be a meal break between the third and the sixth hour relative to the start of shift. Minimum rest period (meal break) of 0.5 hours is required between consecutive duty in a shift.
- The earliest meal break will exist after the completion of the third hour and the latest before the completion of the sixth hour.
- A shift is feasible if each trip is assigned to one duty, and each duty is a sequence of trips that can be performed by a single crew.
- A crew has to start and end (sign on and sign off) its daily shift at the same depot. The spread time is the time elapsed between sign on and sign off in a shift.
- The crews take a break only at relief points and the change over of trains is at the same relief point.

3 MATHEMATICAL PROGRAMMING MODEL

The following is the proposed MIP model for the railway CSP. The objective function is to minimise the total working time of the crew.

3.1 Notations

Indices

i, i'	trips
j	duty
k	shift
ohd	originate at crew home depot
thd	terminate at crew home depot
orp	originate at relief point
trp	terminate at relief point
ots	originate and terminate at any station

Sets

I	set of all trips
I_k	set of trips that can be assigned in shift k ($I_k \subseteq I$)
I_{ohd}	set of trips that originate at crew home depot ($I_{ohd} \subseteq I$)
I_{thd}	set of trips that terminate at crew home depot ($I_{thd} \subseteq I$)
I_{orp}	set of trips that originate at relief point ($I_{orp} \subseteq I$)
I_{trp}	set of trips that terminate at the relief point ($I_{trp} \subseteq I$)
I_{ots}	set of trips that can be sequential in the same duty ($I_{ots} \subseteq I$)
J	set of duties
J_i	set of duties which can contain trip i ($J_i \subseteq J$)
K	set of shifts
K_j	set of shifts for duty j ($K_j \subseteq K$)
CD	set of crew depots ($CD \subseteq RP$)
RP	set of relief points

Parameters

t_{ijk}	driving time of trip i in duty j of shift k
$\zeta_{ii'jk}$	transition time from trip i to trip i' of the j th duty of shift k
$\zeta_{ii'j(j+1)k}$	transition time from trip i of the 1st duty to the trip i' of the 2nd duty of shift k
$\zeta_{ii'k(k+1)}$	transition time from trip i of shift k to trip i' of the next shift
α_{jk}	minimum duration of 1st duty in shift k
α'_{jk}	maximum duration of 1st duty in shift k
$\delta'_{(j+1)k}$	minimum duration of 2nd duty in shift k
$\delta_{(j+1)k}$	maximum duration of 2nd duty in shift k
dT_i	departure time of trip i
aT_i	arrival time of trip i
dS_i	departure station of trip i
aS_i	arrival station of trip i
Wt_{max}	normal working time per shift
Wt_{min}	minimum working time allowed per shift
Wt_k	actual driving time in shift k
St_k	spread time of shift k
St_{max}	maximum spread time allowed per shift

Variables

v_{ijk}	$\in\{0,1\}$	binary variable that assign trip *i* to duty *j* of shift *k*
w_{ijk}	$\in\{0,1\}$	binary variable that denotes the assigning of *i* as the first trip of duty *j* of shift *k*
$x_{ii'jk}$	$\in\{0,1\}$	binary variable that denotes the assigning of *i* is followed by *i'* in duty *j* of shift *k*
y_{ijk}	$\in\{0,1\}$	binary variable that denotes the assigning of *i* as the last trip of duty *j* of shift *k*
$z_{ii'j(j+1)k}$	$\in\{0,1\}$	binary transition variable that denotes the assigning of *i* at the end of duty *j* to be followed by *i'* at the beginning of the subsequent duty of shift *k*
$z_{ii'k(k+1)}$	$\in\{0,1\}$	binary transition variable that denotes the assigning of *i* at the end of the shift to be followed by *i'* at the beginning of the subsequent shift
U	$\in\{0,1\}$	binary variable
$start_{ijk}$	$\in \mathbb{R}$	starting time of trip *i* in duty *j* of shift *k*
$comp_{ijk}$	$\in \mathbb{R}$	completion time of trip *i* in duty *j* of shift *k*

Objective Function

$$\text{Min} \quad \sum_{k=1}^{K} \left(\left| St_k - Wt_k \right| \right) \tag{1}$$

Equation (1) represents the objective function to be minimised. It consists of the sum of the deviation of the total spread time (elapsed time) of the shifts and the actual working time.

Scheduling and Sequencing Constraints

$$\sum_{k \in K_j} \sum_{j \in J_i} v_{ijk} = 1 \qquad \forall\, i \in I \tag{2}$$

Equation (2) is the trip allocation. It enforces every trip *i* to be allocated in exactly one duty *j* of one shift *k*. This constraint accounts only for the shift *k* in which trip *i* is allowed to be scheduled.

$$x_{ii'jk} \geq v_{ijk} + v_{i'jk} - 1 \qquad \forall\; i, i' \in I_k,\, i \neq i',\; j \in J,\; k \in K_j \tag{3}$$

$$z_{ii'j(j+1)k} \geq v_{ijk} + v_{i'(j+1)k} - 1 \qquad \forall\; i, i' \in I_k,\, i \neq i',\; k \in K_j \tag{4}$$

A sequence between trips *i* and *i'* is enforce via constraint (3) when both trips are assigned on the same duty *j*. v_{ijk} and $v_{ii'jk}$ are one when the assignment of trip *i* at duty *j* is followed by trip *i'* at the same duty. Two trips *i* and *i'* are consecutively only in the case that the binary variable $x_{ii'jk} = 1$. Similarly, constraint (4) denotes that the assignment of trip *i* at duty *j* is followed by trip *i'* at the next duty ($j + 1$). The transition variable $z_{ii'j(j+1)}$ is activated when both v_{ijk} and $v_{i'(j+1)k}$ are one. Consequently, one transition from trip *i* to trip *i'* occurs in the end of any duty if and only if trip *i'* is assigned in the subsequent duty.

$$aT_{ijk} \leq dT_{i'jk} \qquad \forall\ i, i' \in I_k, i \neq i', j \in J,\ k \in K_j \qquad (5a)$$
$$aT_{ijk} \leq dT_{i'(j+1)k} \qquad \forall\ j \in J,\ k \in K_j \qquad (5b)$$
$$aT_{ik} \leq dT_{i'(k+1)} \qquad \forall\ i, i' \in I_k, k \in K \qquad (5c)$$

Constraint (5a) ensures that no overlap is allowed. The start time of trip *i'* at any duty requires the completion of the previous trip. Constraint (5b) imposes that the start time of the 2nd duty at every shift *k* requires the completion of the 1st duty. Similarly, constraint (5c) ensures that the next shift will start after the completion of the previous shift.

$$aS_{ijk} = dS_{i'jk} \qquad \forall\ i, i' \in I_k, i \neq i', j \in J,\ k \in K_j \qquad (5d)$$
$$aS_{ijk} = dS_{i'(j+1)k} \qquad \forall\ j \in J,\ k \in K_j \qquad (5e)$$
$$aS_{ik} = dS_{i'(k+1)} \qquad \forall\ i, i' \in I_k, k \in K \qquad (5f)$$

Constraints (5d, 5e, and 5f) are included to ensure connectivity of the trip sequences in a duty.

$$aT_{ijk} \geq dT_{ijk} + \sum_{i \in I_k} t_{ijk}\, v_{ijk} \qquad \forall\ j \in J,\ k \in K_j \qquad (6a)$$
$$start_{i'jk} \geq start_{ijk} + \sum_{i \in I_k} t_{ijk}\, v_{ijk} + \sum_{i,i' \in I_k} \zeta_{ii'jk}\, x_{ii'jk} \qquad \forall\ j \in J,\ k \in K_j \qquad (6b)$$
$$comp_{ijk} \geq start_{ijk} + \sum_{i \in I_k} t_{ijk}\, v_{ijk} \qquad \forall\ j \in J,\ k \in K_j \qquad (6c)$$

Constraint (6a), (6b), and (6c) denote the relation between the start and completion times in a duty. The arrival time of the last trip in a duty is greater than or equal to the departure time of the first trip plus total driving time and total transition time in the duty.

Duty Constraints

The length of time of the dth duty is the sum of all driving times in that duty plus the sum of all transition times in it. Constraint (7a), (7b), (8a), and (8b) express that the length of time of any duty is limited by the total available time for each duty.

$$\sum_{i \in I_k} t_{ijk}\, v_{ijk} + \sum_{i,i' \in I_k} \zeta_{ii'jk}\, x_{ii'jk} \geq \alpha_{jk}\, \mathrm{U} \qquad \forall\ j \in J,\ k \in K_j \qquad (7a)$$
$$\sum_{i \in I_k} t_{ijk}\, v_{ijk} + \sum_{i,i' \in I_k} \zeta_{ii'jk}\, x_{ii'jk} \leq \alpha'_{jk}\, (1 - \mathrm{U}) \qquad \forall\ j \in J,\ k \in K_j \qquad (7b)$$
$$\sum_{i \in I_k} t_{i(j+1)k}\, v_{i(j+1)k} + \sum_{i,i' \in I_k} \zeta_{ii'(j+1)k}\, x_{ii'(j+1)k} \leq \delta_{(j+1)k}\, \mathrm{U} \qquad \forall\ j \in J,\ k \in K_j \qquad (8a)$$
$$\sum_{i \in I_k} t_{i(j+1)k}\, v_{i(j+1)k} + \sum_{i,i' \in I_k} \zeta_{ii'(j+1)k}\, x_{ii'(j+1)k} \geq \delta'_{(j+1)k}\, (1 - \mathrm{U}) \qquad \forall\ j \in J,\ k \in K_j \qquad (8b)$$

Constraints (7a) and (7b) define that the length of time of the first duty *j* should be either greater than or equal α_{jk} or less than or equal α'_{jk}. Whereas constraints (8a) and (8b) define that the second duty (j + 1) also should have a length of time either less than or equal to $\delta_{(j+1)k}$ or greater than or equal to $\delta'_{(j+1)k}$.. This set of constraints enforces that either constraint (7a) and (8a) hold or constraint (7b) and (8b) hold.

$$Wt_k = \sum_{j \in J_i} \sum_{i \in I_k} t_{ijk}\, v_{ijk} + \sum_{j \in J_i} \sum_{i,i' \in I_k} \zeta_{ii'jk}\, x_{ii'jk} \qquad \forall\ j \in J,\ k \in K_j \qquad (9)$$

Equation (9) calculates the total actual driving time in shift k (Wt_k) which is equal to the total working time of all duties in the shift. The total actual driving time within this shift must not exceed the upper bound (Wt_{max}) and lower bound (Wt_{min}).

$$\sum_{j \in J_i} \sum_{i \in I_k} t_{ijk}\, v_{ijk} + \sum_{j \in J_i} \sum_{i,i' \in I_k} \zeta_{ii'jk}\; x_{ii'jk} + \sum_{i,i' \in I_k} \zeta_{ii'j(j+1)k}\; z_{ii'j(j+1)k} \leq St_{max} \qquad \forall\ j \in J,\ k \in K_j \qquad (10)$$

Constraint (10) restricts the spread time of a shift from exceeding the maximum allowed total spread time. Spread time of a shift (St_k) is equal to total working time plus transition time between duties (relief opportunities). The third term in this equation is the transition time between duties in a shift. This transition time includes the time required of the crew to take a meal break during the period of relief opportunities in the shift.

Assignment and Sequencing in a Duty Constraints

Considering that each duty consists of at least one trip, only one trip can be the first or the last one in each duty.

$$\sum_{i \in I_{ohd}} w_{ijk} = 1 \qquad \forall\ j \in J,\ k \in K_j \qquad (11)$$

Equation (11) expresses the requirement that the first trip in the first duty which is also the first trip of the corresponding shift should originate from home depot.

$$\sum_{i \in I_{thd} \cup I_{trp}} y_{ijk} = 1 \qquad \forall\ j \in J,\ k \in K_j \qquad (12)$$

Equation (12) states that the last trip in a duty should terminate at home depot or at relief point.

$$\sum_{i \in I_{ots},\, i \neq i'} x_{ii'jk} = v_{i'jk} - w_{ijk} \qquad \forall\ i' \in I_{ots},\ j \in J, k \in K_j \qquad (13)$$

Equation (13) ensures that each trip, except the first trip, is assigned after another trip.

$$\sum_{i' \in I_{ots},\, i' \neq i} x_{ii'jk} = v_{ijk} - y_{ijk} \qquad \forall\ i \in I_{ots},\ j \in J, k \in K_j \qquad (14)$$

Similarly, equation (14) ensures that each trip, except the last trip, is assigned before another trip.

$$\sum_{i \in I_{thd} \cup I_{trp}} z_{ii'j(j+1)k} = w_{i'(j+1)k} \qquad \forall\ i' \in I_{ohd} \cup I_{orp},\ j \in J, k \in K_j \qquad (15)$$

Equation (15) expresses that for each trip which terminated a duty, there is a transition time (meal break) from this trip to the first trip in the subsequent duty.

$$\sum_{i' \in I_{ohd} \cup I_{orp}} z_{ii'j(j+1)k} = y_{ijk} \qquad \forall\ i \in I_{thd} \cup I_{trp},\ j \in J, k \in K_j \qquad (16)$$

Similarly, equation (16) expresses that for each trip which originated a duty, there is a transition time from the last trip of the previous duty to the current duty.

$$\sum_{i \in I_{thd} \cup I_{trp}} z_{ii'k(k+1)} = w_{i'j(k+1)} \qquad \forall\ i' \in I_{ohd},\ j \in J, k \in K_j \qquad (17)$$

Equation (17) ensures that for each trip which terminated a shift, there is a transition time from this trip to the first trip of the next shift (sign off to sign on).

$$\sum_{i' \in I_{ohd} \cup I_{orp}} z_{ii'k(k+1)} = y_{ijk} \qquad \forall \; i \in I_{thd} \cup I_{trp}, \; j \in J, \; k \in K_j \qquad (18)$$

Similarly, equation (18) ensures that for each trip which originated a shift, there is a transition time from the last trip of the previous shift to the current shift.

4 COMPUTATIONAL RESULTS

Optimisation Programming Language (ILOG OPL Studio) software was used to obtain the solutions. Modeling with ILOG OPL Studio was done by firstly declaring of data and variables. Then was followed by defining the objective function and constraints. Using the CPLEX solver, the results indicate that it is difficult to solve instances using pure MIP method due to the presence of integer variables. Particularly for larger instances, there is difficulty in obtaining feasible solutions because of the large number of variables and constraints. The standard branch and bound technique employed by the CPLEX Mixed Integer Optimiser begins by solving the linear programming relaxation which is obtained by removing the integrality restrictions in the mixed integer program. The number of feasible duties will increase with the number of trips included in the problem as indicated by the proposed technique. When the number of trips increases the CPU times increases as well. The overall results suggest that improvement should be made in terms of search strategy and the time consumed by the methods to obtain the results.

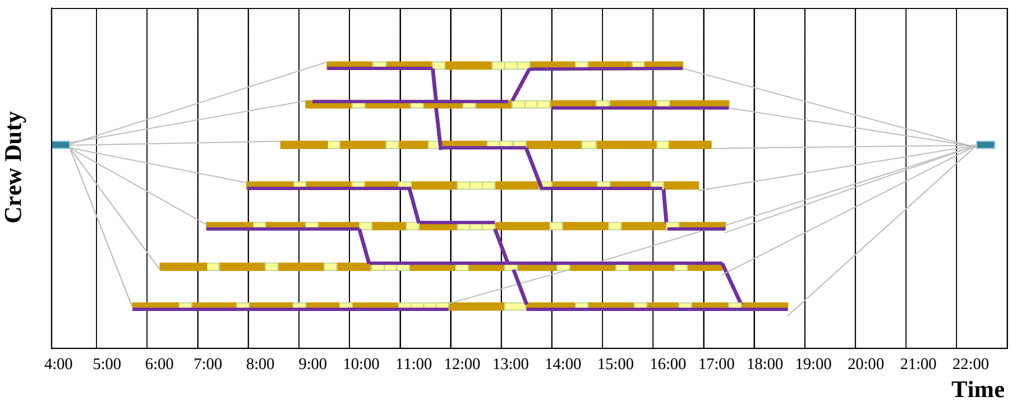


Figure 3. An example of feasible duties

5 CONCLUSIONS

Railway CSP represents a computationally complex problem because of the size of the instances and the complex structure of operational constraints. We present in this work an alternatives formulation for railway CSP. The MIP formulation includes binary variables that determine whether a trip is assigned in a duty and whether it is the first trip, followed by the next trip or it is the last trip. The number of these variables will increase significantly with the number of trips included in the problem. From a practical viewpoint, the solution technique proposed in this work can be integrated with other techniques to find better solutions. Further research to solve this problem should focus on the search techniques for the MIP and integrating the model with other solution methods to combine the strength of the techniques.

REFERENCES

Alfieri, A., L. Kroon, et al. (2007). "Personnel scheduling in a complex logistic system: a railway application case." *Journal of Intelligent Manufacturing* **18(2)**: 223-232.

Bengtsson, L., R. Galia, et al. (2007). “Railway Crew Pairing Optimisation.” *Lecture Notes in Computer Science (LNCS)* **4359**: 126-144. Springer – Verlag, Berlin Heidelberg.

Caprara, A., M. Fischetti, P. Toth, D. Vigo, and P. Guida. (1997). “Algorithms for Railway Crew Management.” *Mathematical Programming* **79(1–3)**: 125–141.

Caprara, A., M. Monaci, and P. Toth. (2001). “A Global Method for Crew Planning in Railway Applications.” In S. Voss and J. Daduna (eds.), *Computer-Aided Scheduling of Public Transport*, *Lecture Notes in Economics and Mathematical Systems*, **505**: 17–36. Springer.

Chu, H. D., E. Gelman, et al. (1997). "Solving large scale crew scheduling problems." *European Journal of Operational Research* **97(2)**: 260-268.

Chu, S. C. K. and E. C. H. Chan (1998). "Crew scheduling of light rail transit in Hong Kong: from modeling to implementation." *Computers & Operations Research* **25(11)**: 887-894.

Elizondo, R., V. Parada, et al. (2010). "An evolutionary and constructive approach to a crew scheduling problem in underground passenger transport." *Journal of Heuristics* **16(4)**: 575-591.

Freling, R., R. Lentink, and M. Odijk. (2001). “Scheduling Train Crews: A Case Study for the Dutch Railways.” In S. Voss and J. Daduna (eds.), *Computer-Aided Scheduling of Public Transport*, *Lecture Notes in Economics and Mathematical Systems*, **505**: 153–166. Springer.

Kwan, R. (2010). "Case studies of successful train crew scheduling optimisation." *Journal of Scheduling*, **14(5):** 423–434

Lezaun, M., G. Perez, et al. (2007). "Rostering in a rail passenger carrier." *Journal of Scheduling* **10(4-5)**: 245-254.

Li, J. (2005). "A Self-Adjusting Algorithm for Driver Scheduling." *Journal of Heuristics* **11(4)**: 351-367.

Sodhi, M. S. and S. Norris (2004). "A Flexible, Fast, and Optimal Modeling Approach Applied to Crew Rostering at London Underground." *Annals of Operations Research* **127(1)**: 259-281.

A LOGISTICS PLANNING METHOD FOR THE TWO-ECHELON DISTRIBUTION SYSTEMS

Jun-Der Leu[1] and Yuan-Kuei Huang[2]

[1,2] *Department of Business Administration, National Central University, Taiwan, leujunder@mgt.ncu.edu.tw*

[2] *Deloitte & Touche Consulting Co., Taiwan, huangyk@csie.nctu.edu.tw*

ABSTRACT

The two-echelon distribution systems have been introduced in many different industries, such as express delivery service companies, e-commerce, hyper-markets products distribution, and so on. For realization of an effective two-echelon distribution system, the logistics execution plays a significant role, wherein the market demand should be fulfilled under given available transportation capacity, besides, the inventory and in-transit materials should be minimized. Since the logistics mechanism of a two-echelon distribution system is quite complex, a computer-based planning tool is normally needed. In this study, we decompose a distribution system into two levels, the first level connects the original depot to the intermediate depots and the second one connects the intermediate depots to the customers. The objective is to maximize the market demand fulfillment considering the time urgency and product value of the demand orders. The developed method is integrated with a googlemap environment and applied to the regional logistics center for a consumer electronics company to validate its solution quality.

Keywords : Two-echelon distribution systems, Logistics planning

1 INTRODUCTION

Distribution in the supply network could be executed by two alternative strategies: direct transportation and multi-echelon distribution (Perboli et al., 2011). In the direct transportation approach, the vehicles starting from a depot and deliver their cargos to their customers, on the other hand in the multi-echelon systems the cargos are delivered from the origin to the customers through intermediate depots. Regional distribution center can be regarded as the space with value-added features in logistics system such as goods gathering, packaging, marking, assembly, processing and export, emergency protection, enhance operational efficiency and etc. They usually located in the integration of production operations, transportation (either by land, sea or air), storage, port and other functions as international logistics zone. International logistics zone are usually adjacent to or including the access to the port with the attached device required for loading and unloading cargo, refueling, space of cargo storage, land and sea freight forwarding and other functions. Furthermore, cargo provides tax exemptions or concessions to avoid customs interference in handling and transporting process (Lu et al., 2008). The globalized enterprises with either production-oriented or market-oriented strategy usually set up regional logistic centers to be responsible for regional goods distribution, assembling process or directly distribute to the demand site.

Marketing channel is defined as the cooperation formed by goods commercial partnership of the existing business members, or a group of mutual aid companies who provide products or services from manufacturers to the end user. Most of the producers transport their products to the market through the cooperation of the market channel (Bowersox & Cooper, 1992; Stern et al., 1996; Kotler, 1994). Among the types of business logistics, the market channel oriented enterprises are not engaged in producing and manufacturing products, but to provide cargo transportation and distribution services. Also, they are likely to set individual regional distribution center for different segments. Enterprises can ship the products to the regional distribution center and customers by the regional distribution center, or launch the final assembly at the regional distribution center before the shipment (Ambrosino & Scutellà, 2005). Therefore, regional distribution center is important especially to trade-based enterprises or those who need to transport the product to various regions.

Most of the logistics planning problems of distribution networks are modeled by the Vehicle Routing Problems in which the solution heuristics are developed (Achuthan etal., 2003; Lapierre et al., 2004; Bräysy and Gendreau, 2005a; Bräysy and Gendreau, 2005b; Mendoza, 2011). In dealing with the coordination problems of production and sales in logistics system, there are a number of scholars have established a variety of different models to reflect the situations of the true world. However, most of the past literature chose to deal with coordination problems with mathematical programming (Flipo and Finke, 2001; Ambrosino and Scutellà, 2005; Tsiakis and Papageorgiou, 2008; Das and Sengupta, 2009), heuristic algorithms (Barbarosoğlu and Özgür, 1999; Jang et al., 2002) and etc. The coordination problems set by the production-oriented enterprises usually focus on the production capacity, system time, and cost as the main consideration without specific solution to the characteristics of market-oriented business.

Therefore, achieving maximum customer satisfaction is the objective of logistics in this study. The logistics management model built with situational parameters according to demand side's urgency and process consideration of product distribution provides routing planning for transporting the final product to selling channels in various regions. To form a feasible method of distribution logistics, we build a computer program environment with the combination of distribution planning algorithm and mapping tools of Google Maps, and then discuss its efficiency through case practice. The second chapter of this study proposed a problem-solving algorithm while Chapter 3 apply and varify the solution of algorithm in Chapter 2 with the actual case of the distribution network in northern Taiwan. Finally, the content of this research is summarized in Chapter 4 with the conclusions.

2 DEVELOPMENT OF THE LOGISTICS PLANNING METHOD

2.1 The planning scenario and solution framework

This distribution method has the following scenario planning assumptions:

1. Assume regional allocation of two layer distribution is known; all the distribution centers and wholesaler belongs to which logistics center and all the selling channels belongs to which distribution centers are also known.
2. There is only one logistics center with known number of distribution centers, wholesalers, and the terminal sales channels but not considering cargo transfer supporting situation between nodes of different region.
3. Assume the distance and path between each node of the distribution network is known.
4. Assume the number of truck teams, driving speed and loading capacity is known.
5. Assume that the transporting network is connected, which means that there is at least one path between any two nodes.
6. After the completion of distribution network planning and shipment, the truck team may pass by the logistics center / distribution center in the distribution path, but would not do planning and loading / unloading operations at this time.

The distribution planning method is decomposed into two levels. The plan of the first level is for the market-oriented enterprises to deliver the final products to distribution centers and wholesalers from the logistics center, named the intermediate depots. The plan of the second level is reaching the target to transport goods to the end customers from the intermediate depots distribution center within the scheduled delivery time. The intermediate depot with its own transportation vehicle team should consider waiting time, distance and total value of the distribution goods as well as having the six main items of situational parameter information as input value to get the distribution path and delivery time of each delivery truck team as the problem solving framework shown in Figure 1.

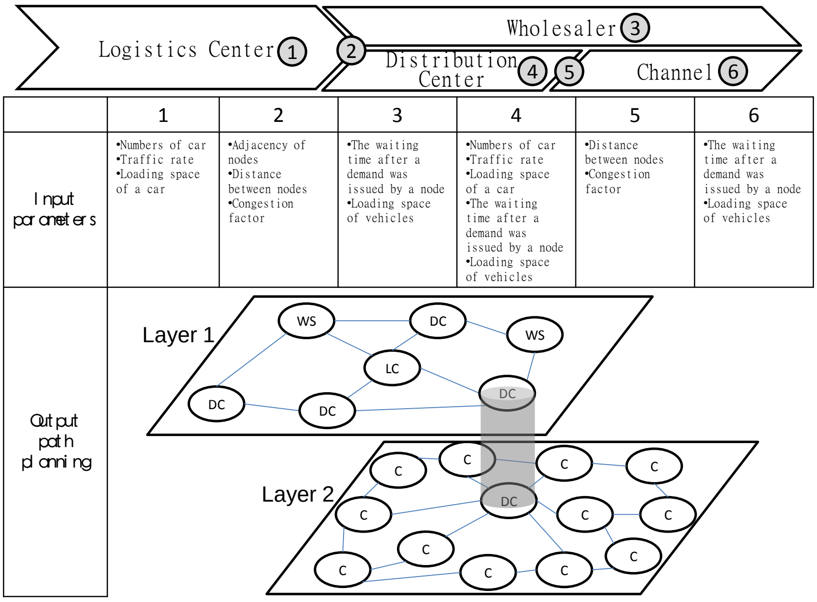


Figure 1. Solution framework of the logistics planning method

2.2 Definition of the emergency factor

In order to meet the needs of each customer and achieve customer satisfaction, the three items of customer waiting time, distance and total value of the distribution items are reflected in the emergency factor, the degree of customer demand (nodes) urgency. The three items that affecting imminence factor are shown as follows:

1) *The waiting time of each node (Waiting time, WT_n)*: the waiting time after a demand was issued by a node. Longer waiting time was regarded the higher degree of urgency.

2) *The value of the demand goods (Value, VAL_n)*: even if the waiting after a demand was issued by a node is short, the node must represent an important customer if the total value of the node is high. Even thought the waiting time is short, the demand should be fulfilled as quickly as possible. Therefore, the higher the demand value, the higher the degree of urgency will be.

3) *Distance (Distance between node m and n, DISmn)*: The farther the node apart from other nodes, the more time and costs of transporting the goods to the node will be. Thus distance can be regarded as a reduction of imminence factor. The farther node away from the current location of the truck team is regarded as less urgent. However, distance parameter means only geographical distance but not the information of terrain and traffic conditions. The required distribution time between the two nodes cannot be accurately reflected. So in addition to distance, congestion (Congestion factor, CF_{mn}) is added to be the basis of distribution time measuring.

Based on the three factors above, two different angles were applied in this study to develop two imminence factor formulas, as follows:

The emergency factor I:

$$IM_{mn} = \left(WT_n - \frac{DIS_{mn} \times CF_{mn}}{VLC} \right) \times VAL_n \tag{1}$$

IM_{mn}: Imminence factor from distribution center m to customer zone n
WT_n : The waiting time after a demand was issued by a node n
DIS_{mn}: The shortest path between nodes m and n
CF_{mn} : Congestion factor between nodes m and n
VAL_n : Total demand value of a node n
VLC : Average traffic rate

Formula (1) shows the first imminence factor measure formula takes waiting time, delivery time, and the total product value as the measuring basis of. When the truck team arrived node m, the imminence factor of the current position of the truck team toward other unsatisfied demands of node n is calculated from the viewpoint of node m to select the most urgent node as the next stop of the distribution team. WT_n is the waiting time after the demand launched from node n with unmet needs. The longer the waiting time of node n, the more urgent would be regarded.

DIS_{mn} is the path distance between nodes n with unmet needs and the current node m; CF_{mn} is the congestion factor calculated based on current traffic conditions and terrain of this path. The actual data of the distance between reaction node m and node n can be obtained from the multiplication of the two. After this distance data divided by the average car speed VLC, we would get the estimated traveling time needed. The longer traveling time represents the higher cost of time and money required, which is regarded less urgent. Therefore, when designing imminence factor measuring formula, we took the traveling time as the reduction that subtracts form node n waiting time WT_n, so as to obtain the degree of urgency measured by the time dimension of the node n. In addition to the degree of urgency measured by the time dimension, the degree of urgency should also consider the value of the node demands. Thus, we multiplied the result of time dimension and the total demand value of the node n VAL_n to obtain node n imminence factor IM_{mn} measured by time, distance, and demand value from node m perspective.

The emergency factor II:

$$IM_{mn} = \frac{VAL_n}{\left(DL - WT_n - \frac{DIS_{mn} \times CF_{mn}}{VLC} \right)} \tag{2}$$

DL: Distribution time windows

Formula (1) regarded the factors of time, distance, and value of the demand as the basis of judging the degree of urgency. Although the measured factors adopted in formula (2) are similar with the one used in formula (1), it designs imminence factor measuring formula with a different angle, and a delivering time limit variables is added. The major design concept of formula (2) is to deduct the waiting time of node n and the transport time from delivery time limit variable DL. The remaining time before the deadline of the delivery can be obtained as the denominator and the total demand value of the node n VAL_n as the numerator. After the division, the imminence factor IM_{mn} of node n from node m can be gained. The more remaining time it is, the greater the denominator would be, and that made urgency factor IM_{mn} smaller and less urgent. Thus, there is more time to deliver the products to the node. On the contrary, it represents more urgency and should be delivered to the node as quickly as possible when there is less remaining time and smaller denominator with greater imminence factor IM_{mn}.

Under certain situations of distribution network, the node imminence factor from formula (2) might be forced to be negative if delivery time limit variable DL is too small, or the total deduction of waiting time and traveling time is too large. Therefore, the denominator of formula (2) would be 1 to make the imminence factor equal the demand value of the node when the situation happens.

2.3 The planning method

Index and parameters

LC	Collection of operations center
DC	Collection of distribution centers
RT	Collection of wholesalers
RD	$RT \cup DC$, the first level of node collection of distribution network
C	Collection of channels
V	Collection of truck teams
U_v	The node with unmet needs on the planned path of truck team v.
$U_{v'}$	the subset of the collection truck team U_v on behalf of the maximize capability of subset collection of the value in truck team v's remaining loading space on the planned path of all the nodes with unmet needs.

Decision variables

D_{rd}	The loading space of the first level distribution network nodes rd's demand amount
D_c	The loading space of the second level distribution network nodes c's demand amount
VAL_{rd}	The total demand value of the first level distribution network node rd's
VAL_c	The total demand value of the second level distribution network node c's
WT_{rd}	The waiting time of the first level distribution network node rd after the demand issued
WT_c	The waiting time of the second level distribution network node c after the demand issued
$DIS_{rd1,\, rd2}$	The distance between node rd1 and rd2 of the first level distribution network
$DIS_{c1,\, c2}$	The distance between node c1 and c2 of the second level distribution network
$CON_{rd1,\, rd2}$	The congestion between node rd1 and rd2 of the first level distribution network
$CON_{c1,\, c2}$	The congestion between node c1 and c2 of the second level distribution network
$IM_{rd1,\, rd2}$	The urgent coefficient of node rd2 from node rd1's perspective of the first level distribution network
$IM_{c1,\, c2}$	The urgent coefficient of node c2 from node c1's perspective of the second level distribution network
VLC	The truck team driving speed
CAP_v	The loading limit of truck team v
$SUM_{v,\, rd}$	The sum of the original loading of truck team v and node rd's loading of the first level distribution network

$SUM_{v,c}$ The sum of the original loading of truck team v and node c's loading of the second level distribution network

2.4 The planning algorithm

Step 1:

Take logistics center (LC) as the starting point of the first level distribution plan. Then, plan the distribution path according to the number of elements of the truck team set (V). If a truck team (v) has sufficient loading capacity when there is existence of node with unmet demands in the distribution network, we calculate the relative emergency factor ($IM_{rd1, rd2}$) from the current position to other node (rd2) base on the position of each truck team (rd1).

Step 2:

Find the greatest values in the current relative emergency factor ($IM_{rd1, rd2}$). Within the truck team (v)'s loading capacity, take the most urgent node (rd2) as the next stop of the delivery route schedule.

Step 3:

Find the shortest path between node (rd2) and the current location of the truck team (rd1) with Dijkstra algorithm. Then, add all the nodes on the path with unmet demands to the unmet demands node set (U_v).

Step 4:

If the remaining loading capacity of truck team (v) is overloaded with the demand of the most urgent node, go to Step 5. Otherwise return to Step 2.

Step 5:

Proceed planning based on the unmet demands node set (U_v) in the distribution path and identify the maximized value of subset (U_v'). If the remaining space (CAP_v - $SUM_{v, Rd}$) of the truck team cannot meet the demand of any node, terminate the path planning process for this truck team. If remaining space of the team is able to meet the demand of more than one node, find out the maximized value of subset (U_v').

The second level distribution program applied the same way to deliver good to the terminal user with the start point changed to distribution center (DC).

3 AN APPLICATION CASE

The developed method is applied to the distribution network of a consumer electronics company in Taiwan as the illustration case. We set the distribution network case in northern Taiwan and established a logistics center with channel-oriented enterprise in Linkou (LC). We set a distribution centers in Taipei City and County, Taoyuan City and County and Hsinchu City and County, and wholesalers in Keelung City (WS1) and Miaoli County (WS2). The distribution center in Taipei area was set in Sanchong district (DC1); the distribution center in Taoyuan area was set in Taoyuan City (DC2); the distribution center in Hsinchu area was set in Hsinchu City (DC3). The distribution from logistics center in northern Taiwan to distribution centers in other cities and counties is taken as a practical example of the first level, and the delivery from the distribution center in Taipei, Hsinchu, and Taoyuan to the end customers is taken as the example of the second level. We use actual distance data while the demand data was gained from random number generator. The study took northern Taiwan in the first level and Taipei City and County as the second level of example. The data of calculation was set as shown in Tables 1 to 5 in the following:

Table 1. Defined parameters of distribution network

Parameter items	Planning Layer

	1st Layer	2nd Layer
Number of nodes	6	13
Number of cars	2	3
Demand value boundary	50 to 100	10 to 40
Waiting time boundary (minutes)	720 to 2,880	100 to 1,440
Loading capacity boundary of a car (units)	20 to 60	10 to 30
Distance boundary (km)	20 to 200	5 to 30
Congestion factor boundary	1.0 to 3.0	1.0 to 3.0
Maximum capacity of a Car (units)	200	100

Table 2. Parameters of the distribution network in northern Taiwan (Distance, Congestion factor)

Nodes	Nodes of 1st layer					
	LC	DC1	DC2	DC3	WS1	WS2
LC		2.1	1.74			1.24
DC1	40				1.69	
DC2	22			2.59	1.31	
DC3			54		1.21	1.84
WS1		25	60	103		1.17
WS2	124			69	161	

Note: Values below the diagonal are distance. Values above the diagonal are congestion factor

Table 3. Parameters of the distribution network in northern Taiwan (Demand value, Waiting time, Consumption of space)

Parameter items	Nodes of 1st layer				
	DC1	DC2	DC3	WS1	WS2
Demand value	58	68	86	75	85
Waiting time (minutes)	1,472	2,158	890	1,345	1,123
Consumption of space (units)	57	56	22	44	27

Table 4. Parameters of the distribution network in Taipei city and county (distance, congestion factor)

Nodes	Nodes of 2nd layer												
	DC1	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
DC1		2.05				2.2	1.13	1.51		1.78		2.9	
C1	13		1.03	2.58									
C2		11			1.31	2.67							
C3		17					2.83						
C4			15			2.12				2.52			
C5	21		27		8				2.73				
C6	17			14				2.68			1.92		
C7	10						18				1.83	2.57	
C8						6				2.58			1.74
C9	16				19				9			1.58	1.23
C10							23	9				2.04	
C11	10							5		8	12		
C12									10	6			

Note: Values below the diagonal are distance. Values above the diagonal are congestion factor

Table 5. Parameters of the distribution network in Taipei city and county (Demand value, Waiting time, Consumption of space)

Parameter items	Nodes of 2nd layer

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Demand value	32	34	27	35	23	13	22	11	35	6	30	33
Waiting time (minutes)	302	1,417	150	723	1,254	170	322	666	1,330	894	228	1,297
Consumption of space (units)	24	17	15	28	15	14	22	27	19	16	17	24

3.1 Validation

In this study, the implementation results of two-level solution with imminence factor algorithm application and case of distribution network were illustrated by program execution result and the time evolve figure. The difference of the order planned with the two formulas was compared. The results of final path planning will be depicted in the real path diagram through Google Maps.

3.2 The planning results

The result shows the distribution order, distribution path, and total traveling distance of each truck team and the arrival time to each node using formula (1). The results of the implementation would be shown on time axis according to the delivery order with different color sector to represent different traveling path of the truck teams as Figure 2. Finally, draw the above-mentioned results of each truck team's traveling path on the map by Google Map as shown in Figure 3 and Fig 4. The distribution order of the map is ranked with alphabetical order.

Two imminence factor measuring formulas were applied in this study to calculate cases of distribution network in northern Taiwan. For formula (2), we set the delivering time limit (DL) to 1,440 minutes. The results was integrated and shown in Table 6.

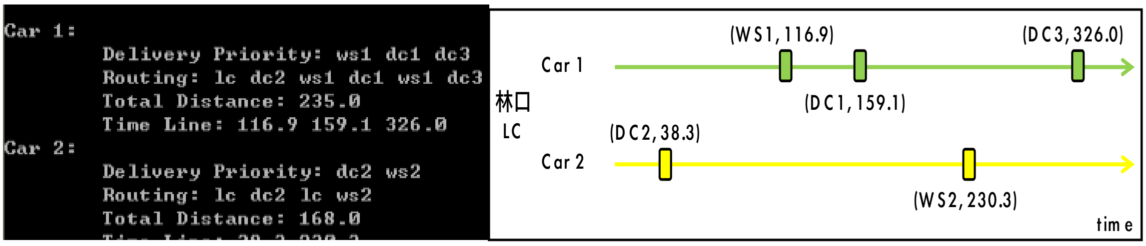


Figure 2. The results of implementation and time evolve figure in northern Taiwan district with using formula (1)

Figure 3. The planning results of transportation route using formula (1)

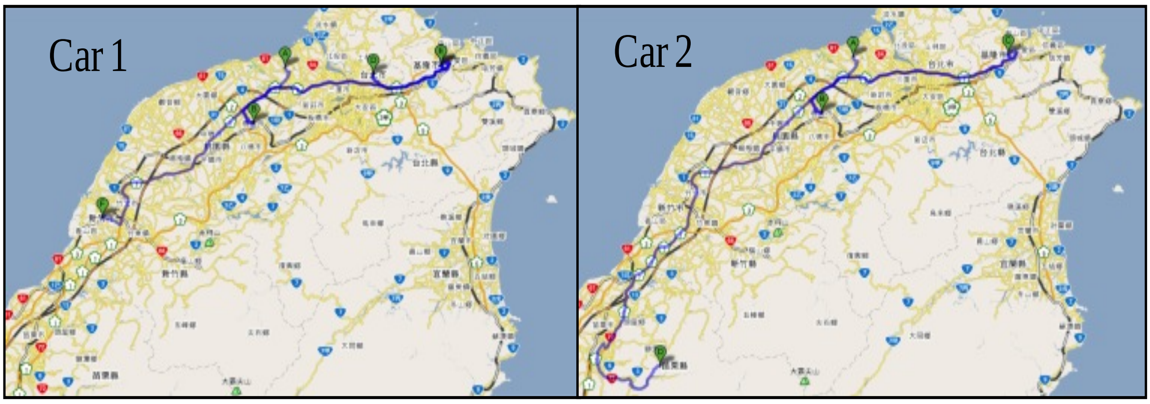


Figure 4. The planning results of vehicle routing using formula (1)

Table 6. A comparison of the results of distribution network

Formula	Performance analysis					
	Car 1		Car 2		Summary	
	Traveling distance (km)	Makespan (minutes)	Traveling distance (km)	Makespan (mininutes)	Total traveling distance (km)	Average makespan (minutes)
(1)	235.0	326.0	168.0	230.3	403.0	278.2
(2)	235.0	326.0	243.0	305.3	478.0	315.6

Note: Default value of DL is 1,440 as using formula (2)

3.3 The execution result of the second level distribution network

The execution results for Taipei area shows the distribution order, distribution path, and total traveling distance of each truck team and the arrival time to each node using imminence factor formula (1). The results of the implementation would be shown on time axis according to the delivery order with different color sector to represent different traveling path of the truck teams. The distribution order of each truck team's traveling path in Taipei district, Taoyuan District and Hsinchu District can also be drawn on the map by Google Map.

4 CONCLUSIONS

This study aimed on the logistics system distribution planning of market channel oriented enterprises who deliver products rapidly to terminal customers through the established logistics centers and distribution centers to meet changes in product demand. We take maximizing customer satisfaction as the point of view with consideration of urgency and product value to develop the distribution planning method for the regional distribution centers. It can be surely applied on practical cases since the imminence factor measuring along with the practical verification of the algorithm's logical correctness proposed by this study. The truck team path finally gained can be applied on practical truck team path planning for delivery.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of National Science Council of Taiwan under grant No. NSC 100-2410-H-008-061. The authors would like to thank the anonymous referees for their valued comments.

REFERENCES

Achuthan, N. R., Caccetta, L. and Hill, S. P. (2003), An improved branch-and-cut algorithm for the capacitated vehicle routing problem, *Transportation Science,* vol. 37, no. 2, 153-169.

Ambrosino, D. and Scutellà, M. G. (2005), Distribution network design: New problems and related models, *European Journal of Operation Research*, vol. 165, 610–624.

Barbarosoğlu, G. and Özgür, D. (1999), Hierarchical design of an integrated production and 2-echelon distribution system, *European Journal of Operation Research*, vol. 118, 464–484.

Bowersox, D. J. and Cooper, M. B. (1992), *Strategic Marketing Channel Management*, 1st edition. McGraw-Hill Companies.

Bräysy, O. and Gendreau, M. (2005), Vehicle routing problem with time windows, part I: Route construction and local search algorithms, *Transportation Science,* vol. 39, no. 1, 104-118.

Bräysy, O. and Gendreau, M. (2005), Vehicle routing problem with time windows, part II: Metaheuristics, *Transportation Science*, vol. 39, no. 1, 119-139.

Das, K., and Sengupta, S. (2009), A hierarchical process industry production–distribution planning model, *International Journal of Production Economics*, vol. 117, 402–419.

Flipo, C. D. and Finke, G. (2001), An integrated model for an industrial production-distribution problem, *IIE Transactions*, vol. 33, 705–715.

Jang, Y. J. and Jang, S. Y. (2002), A combined model of network design and production/distribution planning for a supply network, *Computers & Industrial Engineering*, vol. 43, 263–281.

Kotler, P. (1994), *Marketing Management: Analysis, Planning, Implementation and Control*. 8th edition. Prentice Hall.

Lapierre, S. D., Ruiz, A. B. and Soriano, P. (2004), Designing distribution networks : Formulation and solution heuristics, *Transportation Science*, vol. 38, no. 2, 174-187.

Lu, C.S., Liao, C. H. and Yang, C. C. (2008), Segmenting manufacturers' investment incentive preferences for international logistics zones, *International Journal of Operation & Production Management*, vol. 28, no. 2, 106–129.

Mendoza, J. E., Castanier, B., Guěret, C., Medaglia, A. L. and Velasco, N. (2011), Constructive heuristics for the multicomponent vehicle routing problem with stochastic demand, *Transportation Science,* vol. 45, no. 3, 346-363.

Perboli, G., Tadei, R. and Vigo, D., (2011), The two-echelon capacitated vehicle routing problem: Models and math-based heuristics, *Transportation Science*, vol. 45, no. 3, 364-380.

Stern, L. W., EL-Ansary, A. I. and Coughlan, A. T. (1996), *Marketing Channels,* 5th edition. Prentice Hall.

Tsiakis, P. and Papageorgiou, L. G. (2008), Optimal production allocation and distribution supply chain networks, *International Journal of Production Economics*, vol. 111, 468–483.

HYBRID GENETIC ALGORITHMS FOR THE THREE-DIMENSIONAL MULTIPLE CONTAINER PACKING PROBLEM

Ilkyeong Moon[1], Xuehao Feng[2], Jeongho Shin[3] and Sungchan Kim[1]

[1]*Department of Industrial Engineering, Pusan National University, Busan 609-735, KOREA, ikmoon@pusan.ac.kr,[1] kimsc@pusan.ac.kr[4]*

[2]*Department of Logistics Information Technology, Pusan National University, Busan 609-735, KOREA, fengxuehao@pusan.ac.kr*

[3]*Dongkuk Steel Group, Seoul 100-210, KOREA, jeongho.shin@dongkuk.com*

ABSTRACT

The Three-Dimensional Multiple Container Packing Problem (3DMCPP) aims to pack a set of given finite three-dimensional rectangular items into the minimum number of identical containers without overlapping. The problem's decision framework consists of two main activities: item assignment and packing. This paper presents new hybrid genetic algorithms (HGAs) that address current limitations related to the 3DMCPP and enable a relatively small number of used containers. Specifically, the rotation constraints are also addressed. One HGA is developed for problems that are large in size, and the other HGA is for problems that are small in size. Both of the HGAs combine the Deepest Bottom Left Fill (DBLF) strategy, which is an item assignment strategy, and a basic Genetic Algorithm (GA). Experiments were conducted to demonstrate the performances of the algorithms with two different types of data sets. The results show that the proposed algorithms can achieve fairly small numbers of used containers and within reasonable time limits compared with other algorithms.

Keywords: Multiple container packing, Hybrid genetic algorithm

1 INTRODUCTION

The ocean transportation sector has grown significantly and become an important factor that influences the performances of companies in this sector. To the best of our knowledge, in modern ocean transportation, no choice is used more commonly than containers that are convenient, highly standardized, and relatively inexpensive. It is obvious that with all of the given items packed, a smaller number of used containers can lead to lower container leasing/purchasing and transportation costs. Hence, one of the most common decision problems for managers is how to pack all of the given items into the containers such that the number of used containers is minimized.

The Three-Dimensional Multiple Container Packing Problem (3DMCPP) involves an orthogonal packing of the given items into a minimum number of identical containers without overlapping. Wäscher et al. (2007) proposed a typology and introduced new categorization criteria, wherein the 3DMCPP is described as an input minimization problem. Considering the heterogeneous items, the 3DMCPP is an NP-hard problem, and some algorithms have been proposed for the 3DMCPP. Martello et al. (2000) proposed exact and heuristic algorithms that were shown to be effective when the items were relatively large. The main branching tree is used to assign items to the containers, and a branch-and-bound algorithm is used to decide their actual positions. Faroe et al. (2001) presented a heuristic algorithm based on a guided local search. Starting with an initial number of used containers that were obtained by a greedy heuristic, the algorithm improves the solution iteratively. Lodi et al. (2002) discussed heuristic algorithms for the 3DMCPP. A tabu search approach and the concept of layers are used in the algorithms. Crainic et al. (2009) proposed a two-level tabu search that separates the search for the optimal number of containers from packing the items. The above four papers used the same eight classes of instances, wherein the items are heterogeneous and relatively large. Moreover, in all of the four papers, it is assumed that the items have fixed orientations, i.e., that the items cannot be rotated. On the other hand, Thapatsuwan et al. (2011) proposed three approaches to the 3DMCPP and tested them for heterogeneous and relatively small items. The wall building approach was used in packing the items, and six-way rotation was considered. The experimental results showed that the Artificial Immune System (AIS) can obtain a better solution than the GA and the Particle Swarm Optimization (PSO); however, it needed much longer computational time than the GA. We can observe that the 3DMCPP asks for an orthogonal packing of given items and that filling a single container influences the performances of the algorithms significantly. Ngoi et al. (1994) proposed a spatial matrix approach to record the information of the container and the packed items. Given a sequence of unpacked items, some different packing strategies are used. Bischoff and Ratcliff (1995) used the concept of a layer to pack the items; this approach is widely used in other research on container packing. Karabulut and Inceoglu (2004) investigated a genetic algorithm (GA) wherein the deepest bottom left with fill (DBLF) method is used. Crainic et al. (2008) proposed the extreme point concept for packing items in the container.

The framework of the 3DMCPP contains two activities. In the first activity, the items are assigned to the containers. In the second activity, the assigned items are packed into the containers. Our hybrid genetic algorithms (HGAs) combine the *Largest Left Space First* (LLSF) assignment strategy, the *Deepest Bottom Left Fill* (DBLF) strategy, and a basic genetic algorithm (GA). The GA is used to decide the sequence of items, and the LLSF strategy is used to assign the items to containers. Under the LLSF strategy, each item is assigned to the container with the largest left space. Then, the positions of the items are determined based on the DBLF strategy. The data sets used in Martello et al. (2000) and Thapatsuwan et al. (2011) are used to test our algorithms, and the experimental results demonstrate that our algorithms are competitive.

This paper is organized as follows. The next section describes the 3DMCPP, including a problem definition and a description of the constraints. In Section 3, the proposed algorithms are presented, and Section 4 shows the experiments. Finally, we conclude in Section 5.

2 PROBLEM DESCRIPTIONS AND FILLING A SINGLE CONTAINER

The 3DMCPP can be defined as follows. We are given a set of n three-dimensional rectangular items, and item i is of width w_i, length l_i, and height h_i, $i=(1,2,\ldots,n)$. There are an unlimited number of identical cubic containers with width W, length L, and height H. The objective is to orthogonally pack all of the given items into the minimum number of containers without overlapping. In this paper, we assume that the length, width, and height of the container are oriented with the x, y, and z axes, and six-way rotation is allowed for all of the items. As noted in Martello et al. (2000), it is easy for us to obtain the lower bound for the 3DMCPP. Let V_c be the volume of the container; we can obtain the lower bound LB as follows.

$$LB = \left\lceil \frac{\sum_{i=1}^{n} \left(w_i l_i h_i \right)}{V_c} \right\rceil \tag{1}$$

Martello et al. (2000) discussed another two new lower bounds for the items that are large with respect to the container size, without any rotation. However, in this paper, we still use the *LB* as the lower bound because we consider that both the relatively large and the small items can be rotated in six ways. Figure 1 shows the six-way rotations of an item.

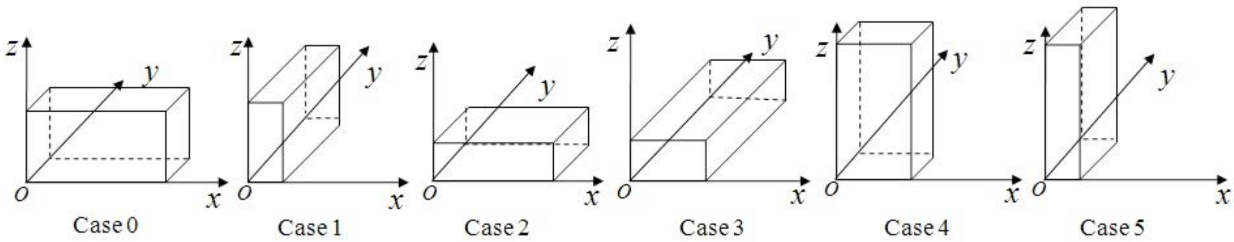


Figure 1. An item in the case of a six-way rotation

Given a sequence of unpacked items and rotations, the question is how to design the packing strategy to obtain a solution of 3DMCPP. Karabulut and Inceoglu (2004) discussed the deepest bottom left with fill (DBLF) method when deciding the position of the items. In our algorithms, the DBLF strategy, in which an item is moved to the deepest available position (the smallest l value) in the layout, then is moved as far as possible to the bottom (the smallest h value) and finally moved as far as possible to the left (the smallest w value), is used to pack a list of items into the containers sequentially.

3 THE HYBRID GENETIC ALGORITHMS FOR 3DMCPP

In the HGAs, the LLSF strategy is used to assign the items to the containers in such a way that the volumes of the items assigned to the containers are balanced. A basic genetic algorithm (GA) is used to determine the positions and rotations of the items based on the DBLF strategy introduced in Section 2.

3.1 The LLSF strategy

Given a set of items, the *LB* is obtained by (1). Then, we generate a list of used containers, and the number of the containers is *LB*. Let *I* be the sequence of the items, and let *C* be the sequence of the used containers. The original number of used containers in *C* is *LB*. The left space of the container is the difference between V_c and the total volume of the items that have been packed in that container. Given an *I*, we assign the items to the containers based on the *Largest Left Space First* (LLSF) strategy as follows.

LLSF strategy:
  begin
    initialize the left spaces of the used containers in *C*;
    for *i*=1 **to** *n* **do**
      select the container with the largest left space;
      if the volume of item *i* is smaller than or equal to the left space **then**
        assign item *i* to that container;
        update the left space of that container;
      else if add a new container into *C* **then**
        assign item *i* to the new container;
        update the left space of the new container;
      end if
    end for
  end.

We can see that at the beginning, all of the containers have the same left space, which is equal to V_c. Then, we assign the largest item to the first container and update the left space of the container.

3.2 The HGA for large size problems

In the large size problem, the number of items is relatively large. We generate a sequence of items based on the volumes of the items following a non-increasing sequence. The assignment of the items follows exactly the LLSF strategy discussed in Section 3.1. Based on the LLSF strategy, the volumes of assigned items of the containers are balanced. Given the assignment of the items, we obtain the chromosomes for each container, which consist of all of the items assigned to the same container. Then, the GA is run separately for each container. In the proposed algorithm, to execute the GA, a chromosome is coded with two rows and m columns, where m is the number of items assigned to the container. Let r_i be the rotation type of item i. As shown in Figure 2, each row presents a different characteristic of the chromosome. The first row presents the sequence of the items, and the second row presents the rotations of the items.

The sequence of the items	1	2	…	m
The rotations of the items	r_1	r_2	…	r_m

Figure 2. The chromosome used in the HGA

As mentioned in Wäscher et al. (2007), the 3DMCPP is an input minimization problem and the following formula is used to calculate the fitness value of chromosome j.

$$Fitness_j = \text{the number of packed items in chromosome } j$$

We can see that the best chromosome has the largest fitness value, which will lead to the smallest number of used containers. Given two old chromosomes for one container, $P1$ and $P2$, both crossover and mutation are used to generate new chromosomes (O) from $P1$ and $P2$. During the crossover that is used, with a probability P_c, two cutting sites i and j are randomly selected, $i<j$. The substring $P1(i)$ … $P1(j)$ is copied into $O1(i)$ … $O1(j)$. Then, $P2$ is swept circularly from j+1 onward to complete $O1$ with the missing nodes. $O2$ can be obtained with the same approach. A two-step mutation is designed for mutating a chromosome, for which two randomly selected sites are swapped with probability P_{m1}, and the rotations of all of the genes of the two new chromosomes are randomly changed with probability P_{m2}. Figure 3 shows an example in which seven items and six-way rotation are considered.

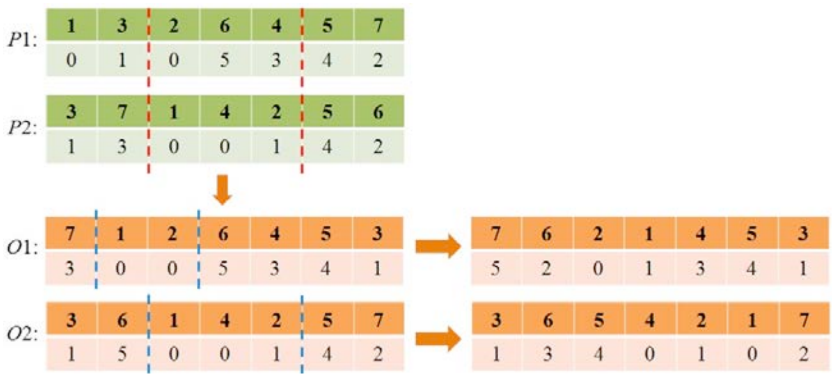


Figure 3. An example of the crossover and mutation

The proposed HGA procedure is illustrated in Figure 4.

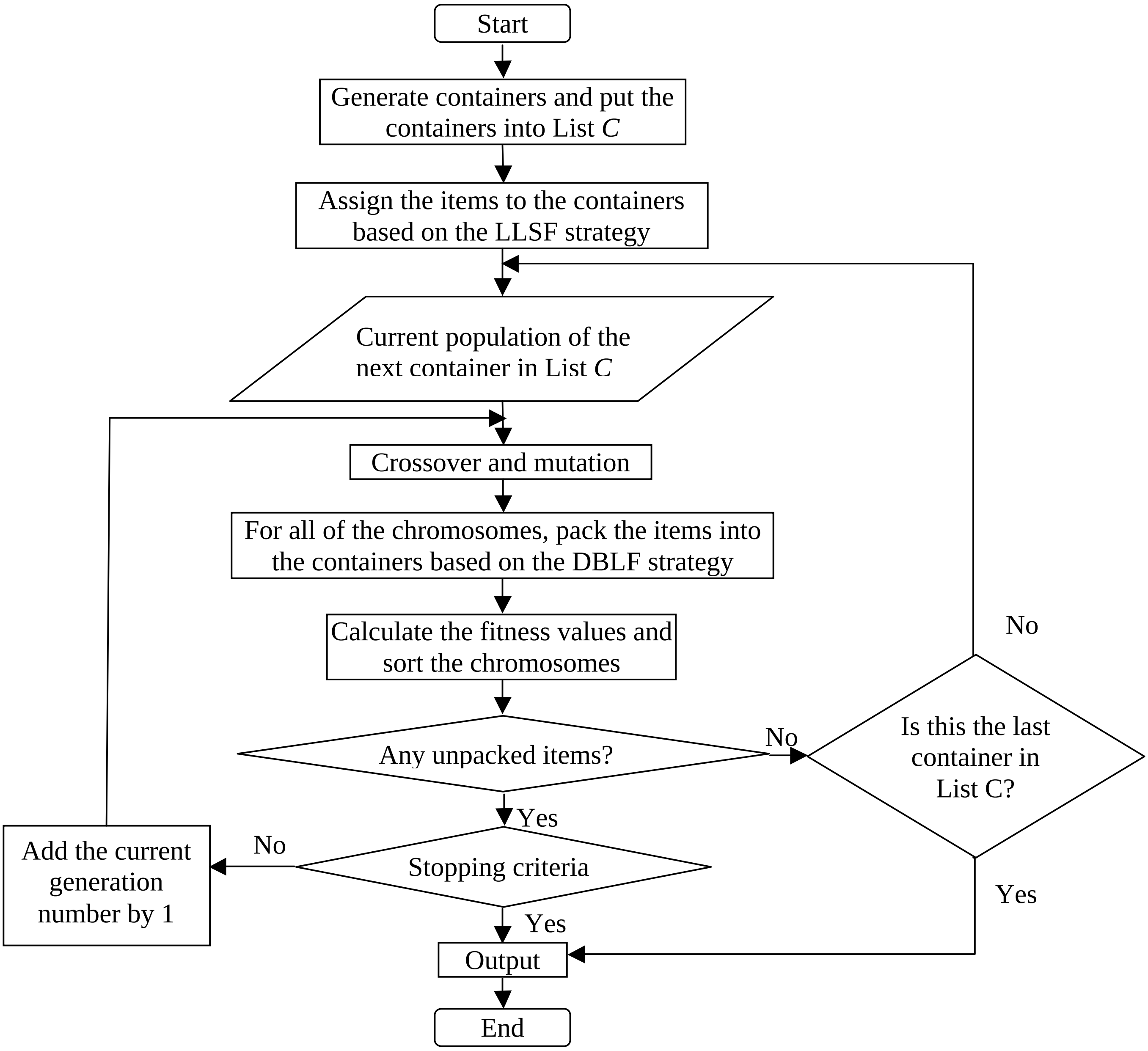


Figure 4. Flowchart of the HGA for large size problems

3.3 The HGA for small size problems

In a small size problem, the number of items is relatively small. In the HGA for the small size problems, the basic GA is used to decide the sequence of the items, and the chromosome has the same structure as the HGA for the large size problems. The differences are that the chromosome here contains all of the items and that the fitness value of the chromosome is the number of used containers. Given a sequence of the items, i.e., a chromosome, we use the basic idea of the LLSF strategy to assign the items to the containers, and we pack the items into the containers based on the DBLF strategy. Given a chromosome, the algorithm for small size problems works as follows.

```
begin
    initialize the left spaces of the used containers in C;
    for i=1 to n do
        select the container with the largest left space;
        if item i can be packed into that container based on the DBLF strategy then
            pack item i into that container based on the DBLF strategy;
            update the left space of that container;
        else
            for j=1 to |C| do
                if item i can be packed into container j based on the DBLF strategy then
                    pack item i into container j based on the DBLF strategy;
                    update the left space of container j;
                    break;
                end if
            end for
            if item i cannot be packed into any container in C then
                add a new empty container in C and pack item i into the new container;
                update the left space of the new container;
            end if
        end if
    end for
end.
```

Because the left space is the absolute value of the left volume, the following case is possible: an item can be packed into a container while it cannot be packed into another container that has a larger left space. Hence, if we fail to pack the item into the container with the largest left space, we also need to attempt to pack it into other containers. Figure 5 illustrates the proposed HGA procedure, wherein the above algorithm is used.

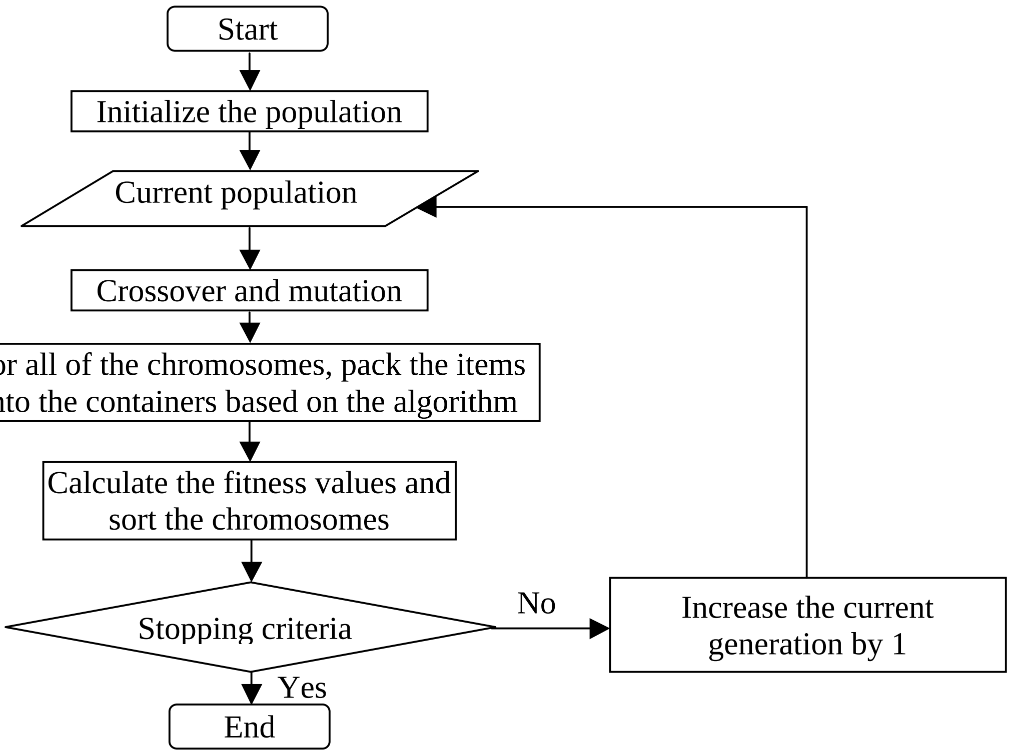


Figure 5. Flowchart of the HGA for small size problems

The HGA for small size problems uses the same structure for the chromosomes, crossover, and mutation mechanisms, which is given in Section 3.2. The assignment of items is determined when the items are packed based on the concept of the LLSF strategy. The assignment mechanism is not exactly the same as the LLSF strategy discussed in Section 3.1. We can observe that the differences between the HGAs that are used in the small size and large size problems are as follows. In the HGA used in large size problems, at the beginning, the items are assigned to the containers based on the LLSF strategy. Then, the GA and DBLF strategies are used for each container separately to determine the sequence, rotations, and positions of the items. Therefore, the fitness value of each chromosome is the number of packed items. On the other hand, the HGA for small size problems is used to decide the sequence and rotations of all the items. Then, the items are assigned to the containers based on the LLSF strategy, and the positions are determined based on the DBLF strategy simultaneously. Thus, the fitness value of each chromosome is the number of used containers. It can be observed that the HGA for small size problems is efficient in obtaining good solutions when the number of items is small because more assignment scenarios are generated. However, it might be inefficient in large size problems because it is difficult for the GA to find a good solution when the number of items in one chromosome is too large. This problem could be more significant when the item rotations are allowed. Hence, we develop two different HGAs for the problems of large size and small size.

4 COMPUTATIONAL EXPERIMENTS

As far as we know, there are two main types of test instances that have been used in related studies. We design two experiments and test our algorithms on instances that are generated with the same approaches used in other studies. The proposed algorithms were coded in JAVA, and the experiments were run on a Pentium 4 3.4GHz processor and 1GB RAM. The population size was set to 100; the probability of crossover P_c was set to 0.8; and the probabilities P_{m1} and P_{m2} in the two-step mutation are set to 0.2 and 0.05, respectively. We denote the instances used in Martello et al. (2000) as small size problems and the instances used in Thapatsuwan et al. (2011) as large size problems.

4.1 Experiment on small size problems

In this experiment, we test our HGA for small size problems, and the maximum allowable number of generations was set to 500. Eight classes of instances were used in the exact algorithm (MPV) proposed by Martello et al. (2000). Because Classes 2 to 4 are similar to Class 1, we test our algorithm only on Classes 1 and 4 to 8. For Classes 1, 4, and 5, the container size is W=L=H=100, and there are five types of items.

Type 1: w_i follows a uniform distribution in [1,1/(2W)], l_i in [2/(3L),L], h_i in [2/(3H),H];
Type 2: w_i follows a uniform distribution in [2/(3W),W], l_i in [1,1/(2L)], h_i in [2/(3H),H];
Type 3: w_i follows a uniform distribution in [2/(3W),W], l_i in [2/(3L),L], h_i in [1,1/(2H)];
Type 4: w_i follows a uniform distribution in [1/(2W),W], l_i in [1/(2L),L], h_i in [1/(2H),H];
Type 5: w_i follows a uniform distribution in [1,1/(2W)], l_i in [1,1/(2L)], h_i in [1,1/(2H)].

For Class k (k=1, 4, and 5), each item is of type k with a probability of 60% and has one of the other four types with a probability of 10% each. Classes 6, 7, and 8 are generated as follows.

Class 6: W=L=H=10, and w_i, l_i, and h_i follow a uniform distribution in [1,10];
Class 7: W=L=H=40, and w_i, l_i, and h_i follow a uniform distribution in [1,35];
Class 8: W=L=H=100, and w_i, l_i, and h_i follow a uniform distribution in [1,100].

Martello et al. (2000) used the average deviation of the lower bound with respect to the solution value Z (number of used containers), and the deviation was defined as $100(Z\text{-}LB)/Z$. Ten instances are generated for each class and size of a problem, and a time limit of 300 seconds is given to the test on each instance. We use the same deviation to evaluate our solutions, and rotation is not allowed. Please note that our HGA can easily add a two-way or six-way rotation. The same time limit of 300 seconds is used. A comparison is shown in Table 1, and n is the number of items. In Classes 1 to 8, the left columns show the results of the MPV, and the right columns show the results of the HGA. From Table 1, we can see that our algorithm obtains better solutions than MPV on most problems. Hence, it can be expected that our algorithm is more competitive on the small size problem when the rotations are allowed.

Table 1. Comparison of solutions on small size problems

n	Class											
	1		4		5		6		7		8	
	MPV	HGA	MPV	HGA	MPV	HGA	MPV	HGA	MPV	HGA	MPV	HGA
10	24.2	20.8	45.5	37.2	40.0	34.2	31.0	29.1	40.0	40.0	28.6	30.6
20	30.0	22.2	46.3	41.8	30.8	26.7	27.8	25.7	34.2	33.8	32.7	22.3
30	30.2	22.9	44.8	41.1	29.6	24.2	20.6	21.2	31.8	29.0	28.3	22.7
40	23.7	23.0	47.3	43.9	31.0	22.6	19.3	22.2	38.1	26.7	26.2	25.9
50	26.6	24.3	46.3	43.8	25.8	24.6	14.3	18.8	35.3	23.4	25.6	24.4

4.2 Experiment on large size problems

In this experiment, we test our HGA for large size problems. We generate our instances with the same approach that was used in Thapatsuwan et al. (2011). A standard container that is 20 ft long, 8 ft wide, and 8 ft high is considered in all of the instances. The length, width, and height of the items follow uniform distributions in [70cm,100cm], [50cm,80cm], and [30cm,60cm], respectively. The maximum allowable number of generations for each container is set to 100. Thapatsuwan et al. (2011) used the waste space to evaluate the algorithms. It can be observed that the objective is the same when minimizing the number of used containers when the containers are identical. Hence, we also use the waste space to evaluate our solutions, and a six-way rotation is allowed (see Figure 1). Fifteen instances are generated for each size problem. As shown in Section 1, the proposed AIS can obtain better solutions than the other two algorithms for all of the problem sizes (see Thapatsuwan et al. 2011). Hence, we compare our results with the results of the AIS that have been presented in Thapatsuwan et al. (2011). The comparison is shown in Table 2.

Table 2. Comparison of solutions in large size problems

n	Average waste space (m^3)		Computation time (min)	
	AIS	HGA	AIS	HGA
100	42.58	12.46	1.26	0.16
250	117.37	48.06	2.95	0.81
500	244.32	60.09	5.78	4.77
750	374.62	108.86	8.35	5.03
1000	510.61	118.95	11.78	9.46
1250	643.57	168.48	15.11	11.22
1500	778.12	181.43	18.63	18.71
1750	912.36	226.65	23.18	18.90
2000	1055.61	241.61	27.30	22.00
2250	1194.65	288.10	31.50	25.67
2500	1332.63	335.73	35.48	38.14

From Table 2, we can see that our algorithm obtains better solutions than the AIS. Note that the volume of each container is approximately 36.25m^3, and we can infer the number of saved containers compared with the AIS. When n=500, for example, the saved average waste space is (244.32-60.09)m^3, i.e., 184.23m^3. Then, we can expect that the solution of the HGA uses approximately 5 containers less than the solution of the AIS. The number of saved containers increases with a larger *n*. In addition, the time that is used to find the solutions of our HGA is similar to the time of the AIS. Thapatsuwan et al. (2011) also tested the AIS on instances in which the numbers of items are from 2750 to 5000. It can be shown that our HGA can obtain better solutions within a reasonable time limit for those instances. In this paper, however, we show a comparison only for the instances from *n*=100 to *n*=2500 for the following reasons: (i) as far as we know, it is not common for a company to pack so many heterogeneous items simultaneously And (ii) before container packing, the managers usually have an original sequence of items. If the packing solution is quite different from the original sequence, it is obvious that the additional cost is also high. This problem might be more significant when the number of items is larger. Hence, experiments in which the number of items is larger than 2500 are not presented in this paper.

5 CONCLUSIONS

We propose hybrid genetic algorithms that consist of the LLSF strategy, the DBLF strategy and a basic GA. An LLSF strategy is used to assign the items to the containers such that the volumes of the assigned items of the containers are balanced. The sequence of the items is generated by a GA, and the DBLF strategy is used to determine the positions of the items. Different HGAs are developed for large size and small size problems, and the experimental results demonstrate the performances of the HGAs. In the HGA used in large size problems, the assignment of items is separated from deciding the positions of the items. At the beginning, the items are assigned to the containers based on the LLSF strategy. Then, the GA and DBLF strategy are used for each container separately to determine the sequence, rotations, and positions of the items. On the other hand, the HGA for small size problems is used to decide the sequence and rotations of all of the items. Then, the items are assigned to the containers based on the LLSF strategy, and the positions are determined based on the DBLF strategy simultaneously.

Compared with the current algorithms in other research, our HGA for small size problems is competitive even when item rotations are not allowed. Considering that item rotations can be easily implemented in our HGA, we infer that our HGA is more competitive in the cases wherein item rotations are allowed. In the large size problems, our HGA can bring significant improvement in the solutions, compared with the current research, and the computational time is acceptable. Our HGA could be improved in several directions. For example, a new heuristic for assigning the items to the containers may increase the effectiveness of the HGA. In addition, designing a new framework of assignment and packing may be another fruitful direction for future research.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0026508).

REFERENCES

Bischoff EE, Ratcliff MSW (1995) Issue in the development of approaches to container loading. *OMEGA International Journal of Management Science,* **23(4)**, 377–390.

Crainic, T. G., Perboli, G. and Tadei, R. (2008), Extreme point-based heuristics for three-dimensional bin pakcing, *INFORMS Journal on Computing*, **20(3)**, 368-384.

Crainic, T. G., Perboli, G. and Tadei, R. (2009), *TS^2PACK*: A two-level tabu search for the three-dimensional bin packing problem, *European Journal of Operational Research*, **195**, 744-760.

Faroe, O., Pisinger, D. and Zachariasen, M. (2003), Guided local search for the three-dimensional bin packing problem, *INFORMS Journal on Computing*, **15(3)**, 267-283.

Karabulut, K. and Inceoglu, M. (2004) A hybrid genetic algorithm for packing in 3D with deepest bottom left with fill method. *Proceedings of ADVIS 2004*, 441–450.

Lodi, A., Pisinger, D. and Vigo, D. (2002), Heuristic algorithms for the three-dimensional bin packing problem, *European Journal of Operational Research*, **141**, 410-420.

Martello, D., Pisinger, D. and Vigo, D. (2000), The three-dimensional bin packing problem, *Operations Research*, **48(2)**, 256-267.

Ngoi, B.K.A., Tay, M.L., and Chua, E.S. (1994), Applying spatial representation techniques to the container packing problem, *International Journal of Production Research*, **32(1)**, 111-123.

Thapatsuwan, P., Pongcharoen, P., Hicks, C., and Chainate, W. (2011), Development of a stochastic optimisation tool for solving the multiple container packing problems, *International Journal of Production Economics*, doi:10.1016/j.ijpe.2011.05.012.

Wäscher, G., Haunβer, H., and Schumann, H. (2007), An improved typology of cutting and packing problems, *European Journal of Operational Research*, **183**, 1109-1130.

A TOOLBASED APPROACH TO ASSESS TECHNOLOGY INTRODUCTION IN TRANSPORTATION SYSTEMS DEMONSTRATED BY THE LNG INTRODUCTION FOR SHIP PROPULSION

Tim Hoerstebrock[1], Christian Denker[1], Tammo Buss[1] and Axel Hahn[1]

[1]*OFFIS – Institute of Information Technology, Escherweg 2, 26131 Oldenburg, tim.hoerstebrock@offis.de*

ABSTRACT

Liquefied Natural Gas (LNG) promises to be an ecological and economical favourable fuel alternative for ship propulsion in comparison to heavy oil. Considering its introduction, port authorities and ship owner face a stalemate due to missing fuelling infrastructure (ship owner's view) or missing modified ships (port's view). Therefore, strategies must be developed that create incentives to make the step towards the new technology. This is a complex task which cannot be solved by statistical approaches since they lack temporal and spatial resolution. In order to assess the introduction of new propulsion systems in existing transportation systems (including its required infrastructure), we propose a multi-agent simulation (MAS) framework. Our approach consists of three layers which include basic functionality of a MAS, concrete agents for traffic systems and a specific implementation of ship models. We demonstrate that all relevant data in different aggregation levels can be integrated in order to assess certain setups of technical ship and infrastructure equipment and to evaluate different LNG introduction strategies.

Keywords: Multi-Agent Simulation, Maritime Transportation Systems, LNG, Technology Assessment

1 INTRODUCTION

The constantly increasing demand of transportation capacity leads to a growing worldwide sea traffic. Most of this traffic is produced by ships using heavy oil as fuel for their propulsion. Heavy oil is a waste product from crude oil refinement, it contains a high proportion of toxic substances like sulphur oxides, nitrogen oxides and carbon dioxide. Reacting to that problem, the International Maritime Organization (IMO) and the European Union (EU) introduced the so-called "Emission Control Areas" (ECA) in 2008. These areas are established in the European sea area and are also planned to be introduced at North America's east coast. Within the ECAs, ships must only use sulphur reduced fuel (1.5% in 2008; 0.1% in 2015).

In order to fulfil IMO's and EU's requirements, ship owners have to switch to more refined fuels such as marine diesel oil (MDO). But this fuel leads to higher costs in comparison to heavy oil which will even rise in the medium term considering oil shortages and the required refinement level of the fuel by 2015. Therefore, ship owners are searching for alternative technologies and/or fuels that are ecologically and economically favourable. Liquefied Natural Gas (LNG) is one of those alternatives. It doesn't contain any sulphur; the nitrogen oxide emissions are 80%, the GHG emissions 20% lower than the emissions from burning MDO (Stuer-Lauridsen *et al.* 2010).

Unfortunately, the usage of LNG as fuel is not possible because ship owners currently restrain themselves from modifying the engines of their ships. They face high investment costs of those modifications, the loss of transportation capacity due to the large LNG tanks and a currently missing LNG bunker infrastructure. Port authorities, on the other hand, do not have economic advantages to install LNG bunker stations because ships are not capable to bunker and to use the provided LNG. This stalemate hinders the market penetration of LNG.

In order to break up this stalemate, strategies must be developed that create incentives for either or both group(s) to make the step towards the new technology. This is a complex task because of the many variables influencing the success of these strategies. These factors include technical specifications of the engines like fuel consumption, fuel properties (costs, emission behaviour, etc.), mobility patterns of the ships and technical specification of the bunker stations. In addition, an overall statistical, average assessment of certain strategies is not practical due to its lack of temporal and spatial resolution. This resolution is needed in order to identify resource bottlenecks and to adapt the evaluated strategy. Therefore, our objective was to develop an assessment tool that is able to evaluate the introduction of new propulsion systems within existing transportation networks including its required infrastructure and its required or dependent systems (e.g. fuel provision logistic).

To solve this problem, we developed a multi-agent simulation (MAS). Since the actors (ship owners, port authorities) of the investigated domain have different sometimes conflicting goals, an agent-based approach fits to represent the stalemate. The core of our approach is an abstract model of transportation systems since maritime traffic is not the only transportation system in which new technology is introduced (e.g. the introduction of electric vehicles for individual transportation). Within our analysis, we simulate the activities of the participating parties over a long time span. In such a traffic domain many different events may be of interest to further analysis which makes a dynamic and deterministic simulation of events and actions inevitable. Section 2 will show that abstract multi agent simulation frameworks fulfilling these requirements are not available.

Each agent within our MAS is connected to a corresponding model which represents the agent-specific data. Ship agents, for examples, have engines a fuel tank and a certain time table. With this, our MAS can derive technical requirements for the ship owner to fulfil their orders, thus making statements about the feasibility to modify the ships with regard to the quality of service. Additionally, new planning algorithms considering the infrastructure layout can be added into the simulation to evaluate their efficiency. From the port’s perspective, fuelling strategies can be assessed that optimize the fuelling plan in terms of throughput and revenue.

The remainder of this paper is structured as follows: section 2 focuses on the related work on MAS and on ecological studies in the maritime sector. We will outline our contribution to these fields of research. Sections 3 and 4 provide an overview about our general approach to assess new technology within transportation systems and will go into more detail by describing our concrete implementation. This will be followed by a short evaluation where we instantiate the concrete objects to enable investigations about the LNG introduction in the North Sea. Section 6 will conclude this paper and will give an outlook about future work.

2 RELATED WORK

In this section we will point out our contribution to the current state of the art in maritime transportation simulation (section 2.1). Section 2.2 will focus on ecological studies in maritime traffic and the feasibility of LNG as fuel for ship propulsion. We will clarify to which extend these studies can be applied to our approach.

2.1 MAS in the maritime sector

There are many different approaches related to the improvement of performance efficiency of traffic systems, transport management and terminal activities. In this chapter, we will concentrate on multi-agent approaches since these approaches are relevant for our research area. To classify these works, we use part of the assessment framework provided by (Davidsson *et al.* 2005). This framework classifies the state of the art by three categories: problem, approach and result. Each category is separated into sub-categories. For this paper, we just concentrate on the problem dimension (see Table 1).

Table 1. Problem categorization of agent-based approaches

Domain	Transport Traffic Terminal	Transport mode	Air Rail Road Sea Intermodal	Time horizon	Operational Tactical Strategical

As we stated in the introduction, our aim is to assess the impact of introducing LNG as a fuel for ship propulsion on ship companies and port/terminal operators. This includes strategic investments in infrastructure and ship equipment. Concerning the assessment framework of (Davidsson *et al.* 2005), our problem consist of all three domains, sea traffic as transportation mode and a strategic time horizon. The result of the research of (Davidsson *et al.* 2005) and our research show that the focus of multi-agent systems and simulation tools concerning maritime traffic mainly face the terminal domain and act on an operational level. They primarily investigate new ways of optimizing the interfaces of the terminal's subsystems berth, storage and road/rail gates. These works focus on the optimal allocation of (transportation) resources (cranes, straddle carriers, trucks)(Kim and Bae 2004; Nishimura *et al.* 2005), synchronizing the operation times and optimizing the berth area, storage area and the container packaging (Moorthy and Teo 2006; Saanen and Dekker 2006a, 2006b; Dekker *et al.* 2007; Kim and Hong 2006; Itmi *et al.* 1995; Kefi *et al.* 2007; Meier 2008). Approaches to solve the (planning) problems differ as well. They include genetic algorithms (Bruzzone and Signorile 1998), petri-nets (Degano *et al.* 2001; Degano and Pellegrino 2002), branch-and-bound algorithms (Kim and Hong 2006) and market-based mechanisms (Henesey *et al.* 2003b). In conclusion, the application of multi-agent technology to support strategic decisions cannot be found.

The problem of strategic decisions, especially for policy makers, is the evaluation of their impact on different stakeholder in the regarded domain. (Henesey *et al.* 2003a) describes an approach to model and simulate different stakeholder relations in a port system. This allows the evaluation of different "what-if" scenarios for decision makers. The port is a highly complex structure with many stakeholders, but only 17% of the ports integrate local communities and stakeholders into their development plan (Brooke 2002). Henesey identifies the different stakeholders of the port system and proposes a MAS-Model based on the MASCommonKADS Framework. This is a good starting point for our work but the focus on the terminal needs to be shifted so that the relationship between ship companies and port operator becomes the centre of investigation. Ji-Wen Dong and Yi-Jun Li also focus the problem of different stakeholders of an intermodal freight transport. They developed a multi-agent system that coordinates the different parties with the aim to reach a seamless intermodal transport (Ji-Wen Dong and Yi-Jun Li 2003).

2.2 Ecological Studies in the Maritime Sector

Since the transportation demand in the maritime sector has increased over the past years, air pollution stemming from ships becomes a growing problem for seaside and landside ecosystems. Matthias *et al.* investigate the effect of ship emissions on the coastal area of the North Sea using a regional chemistry transport model (Matthias *et al.* 2010). Comparing the emissions from heavy oil to those from sulphur-reduced residual oils (2.7% sulphur content), they conclude that sulphur can be reduced by 45% and ammonium aerosols can be reduced by 20%. The integration of the model of Matthias *et al.* in our approach would be useful, since not only the primary effect of the emissions can be evaluated but also the secondary effect on the coastal area. Unfortunately, the authors only focus on MDO and do not evaluate LNG. Derwent *et al.* developed a similar chemistry transport model which evaluates the impact of ship emissions to the coastal areas of the Atlantic Ocean, the North Sea and the Baltic Sea (Derwent *et al.* 2005).

A major contribution concerning the feasibility and evaluation of LNG as a fuel for ship propulsion can be found in (Stuer-Lauridsen *et al.* 2010). Investigating the Danish ferry and short sea cargo sector, they evaluated the benefits and drawbacks of utilizing LNG in shipping and identified the key barriers of LNG. The authors use the scenario technique to evaluate different sets of port/vessel combinations. On this level they aggregate the total amount of consumed LNG fuel and emission savings.

Corbett evaluates the emissions from shipping with a mathematical approach (Corbett 1997). Three sources provided the data for the investigation: marine exhaust emission test data, international marine-fuel usage information and engine characteristics of the ships.

Bengtsson investigates the ecological impact of changing to alternative fuels (Bengtsson 2011). The author takes the whole lifecycles of that switch into account. She compared several fossil fuels like heavy oil, marine diesel oil, LNG and synthetic diesel including the use of additional technology like scrubbers. In addition, two possible roadmaps were investigated to meet the ecological requirements of the IMO: one by using marine diesel oil and one by using LNG. The author concludes that LNG roadmap promises a better overall environmental performance.

In conclusion, the studies about ecological studies in the maritime sector do neither provide technological recommendation nor support ship owners or port authorities in their decisions. However, deal with an overall assessment of the new technology in terms of ecological efficiency and feasibility. Therefore, these studies can serve as data sources for our assessment of the new technology.

3 APPROACH

As stated above, not only the maritime sector but also other transportation domains face strategic questions when it comes to adapting new technology. Therefore, our approach needs to cover all types of transportation systems (and their combination) like car, railway or air traffic.

The challenge of our approach was to set up a highly generalized abstract data model which can be extended to fit (almost) every research purpose in the context of introducing new propulsion technology within existing transportation systems. The researcher (user) is able to derive vehicles (e.g. ship, train, car, airplane), transhipment places (e.g. bunkers, harbours, gas stations, airports) and domain specific route networks. The goal is to be as specific as needed concerning the domain problem, but generic enough to be adaptable to future changes inside/outside of the domain.

Our approach incorporates the use of a MAS. Technically, the simulation is discrete event driven. The reason for choosing a discrete event simulation was to simulate only necessary time slots. This allows the user to efficiently simulate large timespans without wasting any computing power for simulating unnecessary simulation time. Additionally, we are confronted with a system which highly fits to the characteristics of an ideal application for multi-agent technology defined by (Parunak 1999): modular, decentralised, changeable, ill-structured and complex. Thus, we enabled the agents to simultaneously act and designed behaviour dependent communication between these agents, which might invoke further actions. When combing discreteness and communication between autonomous agents, it is essential to care about altering discrete time slots. For instance, while fuelling, a ship agent might prolong the duration. Such an alteration would also cause the corresponding bunker to change fuel delivery plans due to the extended time span. Hence, alternation of the discrete time slots needs to be cascaded throughout every connected agent and its behaviours.

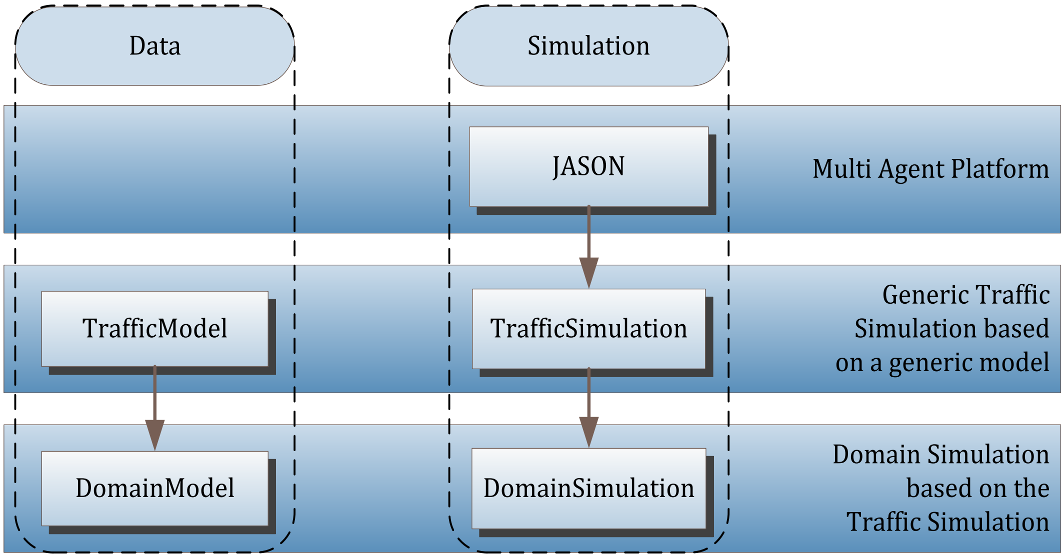


Figure 1. Approach overview

The various agents can follow any goal and therefore need to be customized to focus a specific user defined research purpose. The architecture consists of three levels (see Figure 1). The first level is the abstract implementation of a multi agent simulation platform called JASON. It contains a basic agent which can handle different behaviours simultaneously, a discrete scheduler for actions of various agents in parallel and the possibility to establish communication between agents (see section 4.1). The traffic simulation and the traffic model are positioned on the second layer. The traffic model was built to reflect all necessary abstract objects of the reality. These objects are any types of vehicles, stationary loading units and a network. Each vehicle and stationary loading unit is applied to an agent. This assignment serves as a connection of the traffic model to the traffic simulation. The traffic simulation is a derivate/extension of the JASON package. It describes how vehicles are able to move on the network and how they can load and unload any type of fuel or cargo. In order to receive a tool for the concrete research purpose, three steps need to be done: (1) derive a specific data model from the traffic model, (2) derive a specific simulation model from the traffic simulation and (3) link both derivations with each other. These steps lead to the third layer including the domain model and the domain simulation.

The next section deals with the concrete implementation of the three layers including their corresponding data and simulation models.

4 IMPLEMENTATION

We separated our implementation in two major tracks: model and simulation (see Figure 2). The traffic model within the model track was created using the Eclipse Modeling Framework (EMF) with its Ecore UML-dialect. The advantage of EMF within this approach is the model driven development which implies automatic code generation of the model classes and the model editor classes. The generated editor enables the user to create or edit instances of the domain model over a graphical user interface and thus makes the creation of models over program code obsolete. This allows the user to easily setup his scenarios for evaluation.

Since the specific simulation mainly includes simulation logic, there is no advantage to use the EMF to also support the simulation track. The domain specific simulation packages (domain simulation) are, therefore, specialized in the conventional programmatic way. Further details about the concrete implementation are provided in the following sections.

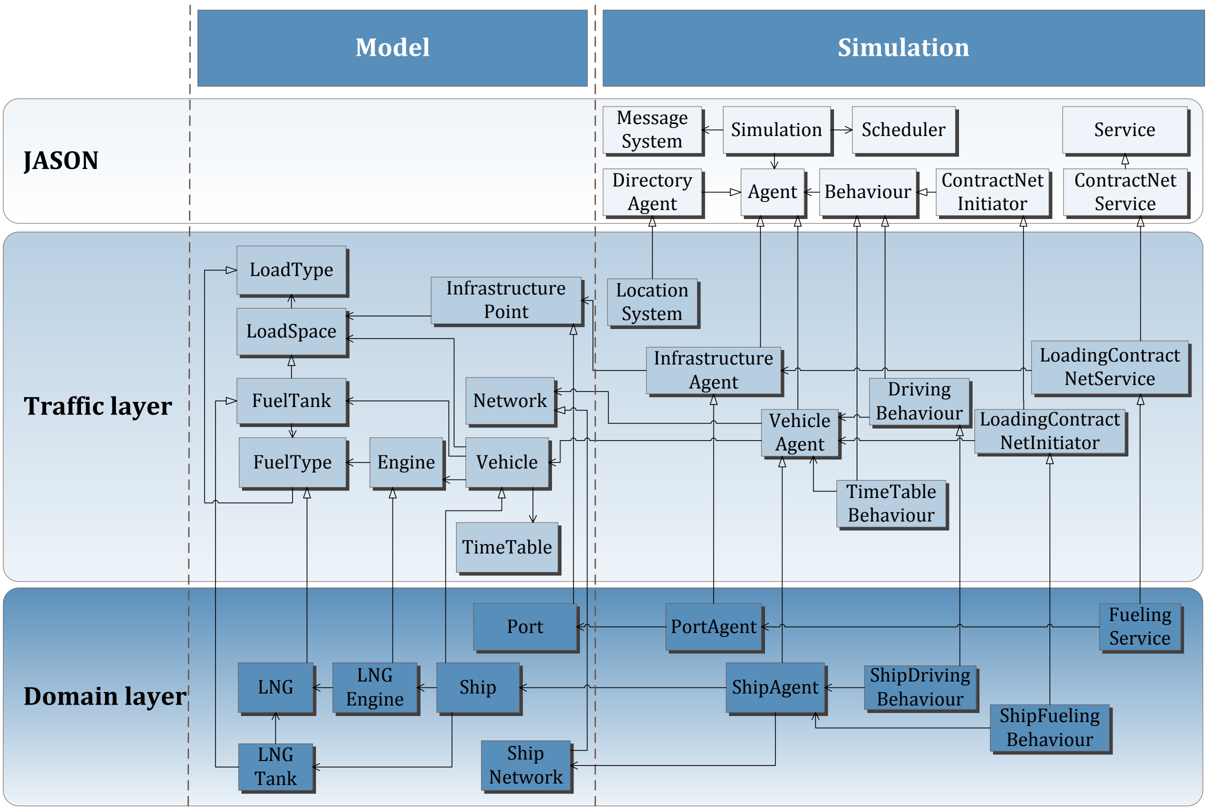


Figure 2. Design of the three layers

4.1 JASON – Java Agent Simulation OldeNburg

The basic simulation framework JASON provides the necessary functionalities of a discrete and event based multi-agent simulation (see Figure 3). The AGENT class is the basic superclass for user defined agents. Agents act inside a simulation and can interact through message based communication. Every agent has an agent description with its unique name and supported services, e.g. fuelling. An agent acts through behaviours that are scheduled by the agent to perform actions at a specific simulation time. The agent provides basic functionality for its behaviours, e.g. sending and receiving messages, scheduling and un-scheduling of behaviours and the ability to add or remove provided services.

The class SIMULATION is the core of the Framework. All agents have to register with it. Registered agents are then able to schedule their behaviours into the SCHEDULER at a specific simulation time. Since the simulation is discrete, only time steps with scheduled behaviours will be executed. Thus, the simulation does not need to simulate every time step but only steps, when behaviours need to act.

The BEHAVIOUR class is the basic super-class for user defined behaviours. Behaviour contains the logic of specific actions an agent can perform. They can schedule themselves or other behaviours within the agent. As long as an action is performed, it is guaranteed, that the current simulation time will not continue. Behaviours can perform actions that last over a specific period of simulation time by rescheduling themselves. The SCHEDULER always calls the action method of a behaviour; therefore a behaviour has to remember its own status and the corresponding tasks that are involved. Due to an agent's lock, it is guaranteed, that only one of the agent's behaviours runs at a time.

Communication is performed through messages sent and received by the agent. For the communication of agents a dedicated message system is provided. All agents registered with the simulation can be addressed. The agents provide an interface for behaviours to send messages through the agent. Thus, all behaviours are able to communicate in parallel. If an agent sends a message with a valid receiver, it will get an answer in any case, at least a NOTUNDERSTOOD message. The simulation framework itself provides basic communication protocols in accordance with the FIPA protocols. Therefore messages are designed aligned to the FIPA ACL Message Standard (Dale 2012).

The implementation of the protocols provides basic functionality like creation of pre-designed messages. Users only have to set certain parameters to create a valid message inside a protocol. The main reason for implementing pre-designed protocols is to guarantee easy negotiations between agents, especially with 1:n relations. Already included in the framework are the contract net, subscribe, and register protocols. Every protocol is based on the initiator and responder classes which provide the necessary interfaces. This design allows a logical and easy to understand disjunction of the participating parties in the protocol.

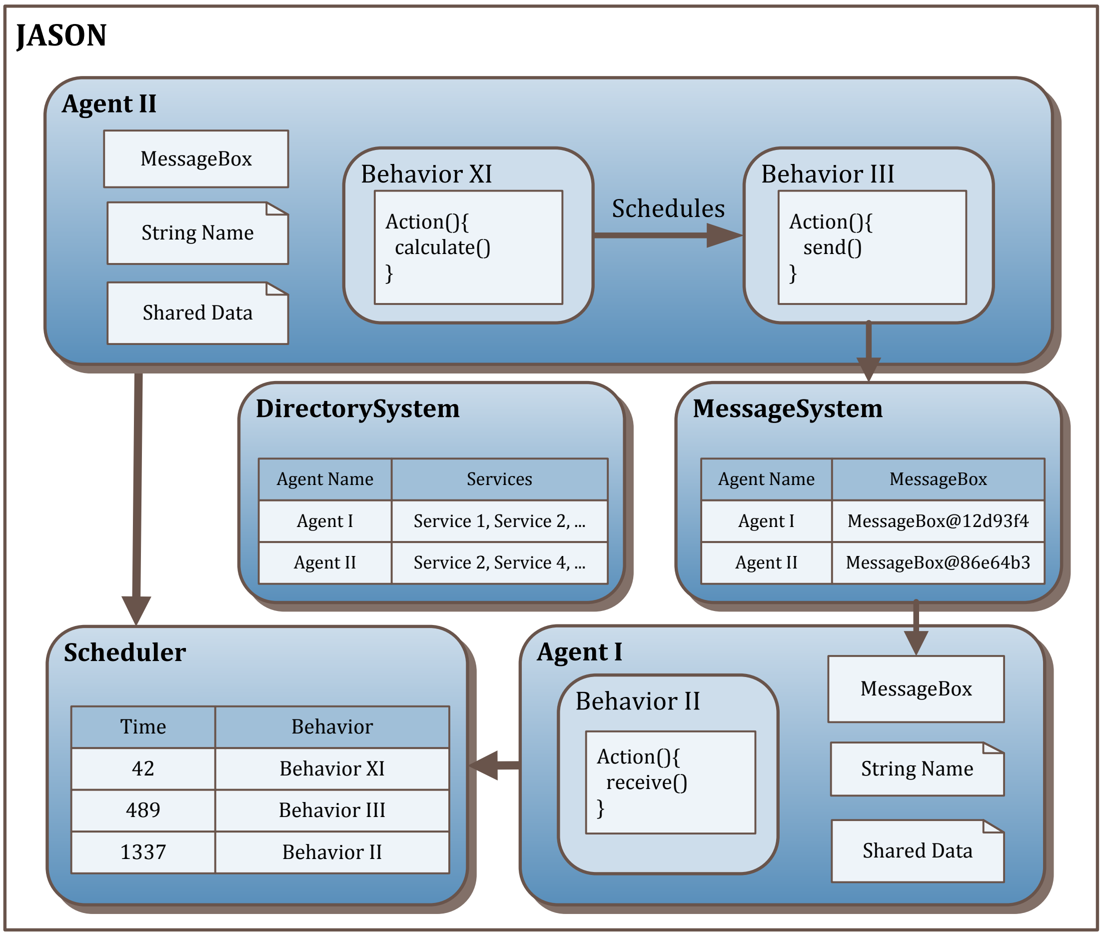


Figure 3. Abstract design of JASON

Agents supporting different services need to register them with a Directory System Agent, which is acting as yellow pages. With this agent, other agents are enabled to search for supported services. The agents' services are registered with the system in an agent description, containing all supported services and the agent's name, to contact it directly if a service should be called on.

4.2 Traffic layer

The traffic layer consists of two parts: traffic model and traffic simulation. The traffic model contains the general elements of traffic system like vehicles and infrastructure. They include technical specifications like capacity, loading/unloading patterns, maximum power of the engine, current and maximum speed. The traffic simulation covers the abstract agents and behaviours that are necessary to represent a transportation system with JASON. In the following, the traffic model and the traffic simulation will be explained in more detail.

4.2.1 Traffic simulation

In an effort to create a generic simulation framework an extension of JASON has been implemented. This extension enables the simulation of all kinds of transportation systems with only few extensions needed. For that purpose, the basic agents, behaviours and services from JASON are extended to represent the basic units of a transportation system. The two main agents are INFRASTRUCTUREAGENT and VEHICLEAGENT. The INFRASTRUCTUREAGENT represents an entity which provides services to other agents in the system. In our case, this service is mainly loading and unloading some sort of goods. This function is covered by the LOADINGCONTRACTNETSERVICE class which extends the basic CONTRACTNETSERVICE of JASON. The INFRASTRUCTUREAGENT registers the service to the LOCATIONSYSTEM which extends the DIRECTORYAGENT. This system allows other agents to find services on the network (e.g. gas stations).

The VEHICLEAGENT is an entity which moves within the transportation system to pursue certain targets. They have certain behaviours to reach these goals. The three main behaviours in the traffic layer are TIMETABLEBEHAVIOUR, DRIVINGBEHAVIOUR and LOADINGCONTRACTNETINITIATOR. The TIMETABLEBEHAVIOUR is supposed to be a part of the inner goal of an agent. It holds the information about the next steps an agent should perform. The basic goal in this abstract layer is to simply execute all defined action on schedule. The DRIVINGBEHAVIOUR is supposed to accomplish the driving on a network. For that purpose, it will check the generic cost of a connection. Within the concrete implementation, these costs can reflect, for instance emissions, fuel or time. The costs are calculated by a Dijkstra algorithm to efficiently find the cheapest route. If enough energy is available, the DRIVINGBEHAVIOUR will start driving and reschedule when the route will be finished. While an agent is driving, it is still able to receive messages and react on events. The LOADINGCONTRACTNETINITIATOR behaviour initiates a negotiation between the VEHICLEAGENT and the INFRASTRUCTUREAGENT which provides the appropriate LOADINGCONTRACTNETSERVICE. After the two parties agreed on price and amount of the required good(s), the loading/unloading takes place.

4.2.2 Traffic model

The traffic simulation from above provides the needed infrastructure to simulate the basic events in a transportation system. The traffic model complements this infrastructure by a suitable data structure to represent the agents' technical specifications. This model is needed to compare different sets of equipment and, therefore, to assess new technology in existing transportation systems. The main agents of the traffic simulation (INFRASTRUCTUREAGENT and VEHICLEAGENT) are connected to their corresponding data models (VEHICLE and INFRASTRUCTUREPOINT).

The technical elements of the VEHICLE cover its ENGINE(s), the corresponding FUELTYPE(s) and one or more additional LOADSPACE(s) if necessary. The ENGINE(s) contain several specifications about efficiency, maximum power, torque, etc. Every ENGINE is bound to one specific FUELTYPE. Hence, programmatically it is prevented from consuming fuels which it is not designed for (e.g. LNG in a MDO engine). The FUELTYPE provides criteria to evaluate its ecological and technological influence like CO_2, NO_x, SO_x, and energy content. In order to represent logistic scenarios, additional LOADSPACE(s) can be assigned to the VEHICLE to represent its transport capacity for other cargo. Each LOADSPACE can store one or more LOADTYPES each representing a certain type of cargo (e.g. containers). These LOADSPACES can be limited by amount and weight.

On the top level, the VEHICLE provides abstract methods to calculate the overall fuel consumption over a defined timespan and to calculate the optimal power input with respect to fuel efficiency. It also contains strategies about the use of its engine(s). This enables the user to define how the engine(s) should be handled and used during simulation. In case of two engines, for instance, one engine could be used as main engine while the second engine is only used within defined situations like dealing with strong headwinds.

As stated in the traffic simulation, the VEHICLE has a TIMETABLEBEHAVIOUR to represent its inner goals. Therefore, a concrete TIMETABLE is provided to specify the destinations, arrival times and the actions which need to be performed at the destinations.

In order to specify which transportation system a VEHICLEAGENT (traffic simulation) can use, the traffic model holds one or more networks to represent roads, railways or seaways. These networks are directly assigned to the VEHICLEAGENT.

The INFRASTRUCTUREPOINT simply has one or more LOADSPACES with certain LOADTYPE(s). It manages these LOADSPACE(s) and holds the data about possible loading/unloading performance and certain restrictions about the LOADSPACE(s). It is assigned to one node of one of the networks of the traffic model.

4.3 Domain layer

The traffic simulation is not concrete enough to investigate a specific transportation domain. Therefore, a domain simulation is needed which is the final extension. At this point, all abstract classes of the traffic simulation are implemented which leads to a runnable simulation tool. Additionally, domain specific data models are created by extending the traffic model to their final classes.

4.3.1 Domain simulation

In order to represent our evaluation scenario, the basic functions from the traffic simulation are sufficient. Since the agent classes are abstract and, hence, do not work out of the box, they are extended by the SHIPAGENT (extends VEHICLEAGENT) and PORTAGENT (extends INFRASTRUCTUREAGENT). They are complemented by certain setup routines to initiate the simulation.

4.3.2 Domain model

Just like the traffic simulation, the traffic model needs another extension to represent the specific technical attributes of the ships involved in our evaluation scenario. This extension can be done by programmatically or model-driven derivation. The model-driven approach includes the utilization of the Eclipse Modeling Framework. The user is forced to implement abstract methods to achieve the required domain specific functionality. This implementation can simply be a taking over the basic functionality of the traffic model or by extending it to fit to the user's needs. The fuel consumption behaviour of an engine, for example, can be modelled as a either linear function only depending on its current power or a multi-criteria function depending on load weight, ship design, wind direction and wave course.

In order to constitute a runnable domain model, the user must extend at least five abstract classes of the traffic model: VEHICLE, ENGINE, NETWORK (plus the corresponding node class) and at least one action for the TIMETABLE. Additionally, all other classes may be extended. An extension of INFRASTRUCTUREPOINT, for example, could be the need to model and simulate a jack-up rig which has to handle its geographical position and storage facilities in a certain way. After extension, these constructs can be seamlessly integrated into the domain simulation.

5 EVALUATION

To evaluate our approach, a ship simulation was created considering a real world scenario. The data used to model the scenario was provided by a regional ferry operator. That data includes information about its fleet, like ship's and engine's specifications, concrete timetables and the corresponding routes used by its ships. In order to achieve a working simulation of the scenario the three steps to construct a ship-layer were performed.

First, the ship model was derived from traffic model (see 4.3). Within this step, we added different fields like *maxSpeed* to the SHIP in order to be able to insert several restrictions like limiting the driving speed of a ship. After derivation, we implemented the abstract methods. For instance, the ship's engines energy conversion efficiency curves were implemented to reflect accurate consumption of each engine. Since many of the fleet's ships use multiple engines to drive, the ships methods to control and distribute the needed power consumption to the engines were specified. Another challenge was to model the fact that the ferry operator's passenger ships drive in the wadden sea area. This area has speed limitations and belongs to an ECA. Therefore, fields were added to the ship network, to enable editing of the restrictions and to incorporate them within the simulation. As a model for load, fuel and ships' timetables the corresponding basic classes of traffic model could be used with no further customization.

Second, the ship simulation was derived from traffic simulation. This included the derivation of a SHIPAGENT from the traffic simulation's VEHICLEAGENT and the derivation of a PORTAGENT from the traffic simulation's INFRASTRUCTUREAGENT. The SHIPAGENT was equipped with a few behaviours. For instance the traffic simulation's DRIVINGBEHAVIOUR was extended to drive according to the ferry operator's timetables using the provided TIMETABLE classes. A functionality to log fuel consumption was also integrated in the DRIVINGBEHAVIOUR. Another behaviour was constituted which controls the ships' fuel tank fill level and initiates a SHIPFUELINGBEHAVIOUR at a certain low fill level. To keep it simple, this is done while the ship is in a port.

If the port does not support the fuel needed, the SHIPFUELINGBEHAVIOUR locates the nearest port, which is on the ship's route and offers the ships' fuel and postpones fuelling to arrival in the next port.

Furthermore, the SHIPFUELINGBEHAVIOUR is a derivation of the LOADINGCONTRACTNETBEHAVIOUR which is used to negotiate fuelling with PORTAGENTS located in a port. Therefore, the PORTAGENT offers a FUELINGSERVICE which is a derivation of LOADINGCONTRACTNETSERVICE.

Finally, to feed the real data into the simulation an EMF Editor was created out of the ship model. The editor was used to build instances of the ships, ports, timetables and the network in the scenario. It was also used to link the SHIPAGENTS and PORTAGENTS with these instances. A simulation could be run that generates realistic data.

The evaluation validated the functionality and is a proof of concept of our approach. The three layer model enables simple building of a traffic based simulation. With JASON and the traffic layer given, researchers are able to quickly build a domain layer examine certain research topics within the field of traffic. The decisive factor is that abstraction and generalization enhances the usability of the shown framework.

6 CONCLUSION AND OUTLOOK

In this paper, we have demonstrated an approach to assess the introduction of new propulsion systems in existing transportation systems with a model-driven multi-agent based simulation.

Based upon a three layered system it is possible to assess certain setups of technical equipment of ships and infrastructures and therefore to evaluate different LNG propulsion introduction strategies.

Furthermore, a toolkit for simulation based analysis of different possible solutions to a traffic based problem is provided. Due to the problem's complexity the appliance of the framework to specific domains requires programming experience and (usually confident) domain specific knowledge including the access to extensive data sources.

All in all, this approach offers the possibility to evaluate problems in a more holistic way. Future work will focus on simplifying the derivation of domain-specific models by tool support so that the programming experience of the user is not a key factor to use the framework for his questions. Parallel, we plan to build a complete scenario of the North Sea ship traffic, so that not only ferry operations can be assessed. Finally, it is even possible to expand the system boundaries over different domains. This could include the evaluation of the street or rail based supply chain of the ports. The framework is flexible enough to also integrate different traffic systems into one simulation.

REFERENCES

Bengtsson, S. (2011). *Life Cycle Assessment of Present and Future Marine Fuels*. Chalmers University of Technology, Gothenburg, Sweden.

Brooke, J. (2002), Stakeholder Participation in Port Development Decision Making: How?, Where? and When? In *Proceedings of the 30th International Navigation Congress (PIANC 2002)*. Sydney, Australia, 1990–1991.

Bruzzone, A. and Signorile, R. (1998), Simulation and Genetic Algorithms for Ship Planning and Shipyard Layout. *SIMULATION*, doi: 10.1177/003754979807100202.

Corbett, J. J. (1997), Emissions from Ships. *Science*, doi: 10.1126/science.278.5339.823.

Dale, J. (2012), Welcome to the Foundation for Intelligent Physical Agents. http://www.fipa.org/index.html. Accessed 14 February 2012.

Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J. and Wernstedt, F. (2005), An analysis of agent-based approaches to transport logistics. *Transportation Research Part C: Emerging Technologies*, doi: 10.1016/j.trc.2005.07.002.

Degano, C. and Pellegrino, A. (2002), Multi-agent coordination and collaboration for control and optimization strategies in an intermodal container terminal. In *Managing technology for the new economy*: *St John's College, Cambridge, UK, 18 - 20 August 2002 ; proceedings*. Piscataway, NJ: IEEE Service Center, 590–595.

Degano, C., Di Febbraro, A. and Fornara, P. (2001), Fault diagnosis in an intermodal container terminal. In *Proceedings*: *ETFA 2001, 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation : October 15 - 18, 2001, Antibes - Juan les Pins, France*. Piscataway, NJ: Institute of Electrical and Electronics Engineers Customer Service Dep, 433–440.

Dekker, R., Voogd, P. and Asperen, E. (2007), Advanced methods for container stacking: Container Terminals and Cargo Systems. In K. H. Kim & H.-O. Günther (Eds.) : Springer Berlin Heidelberg, 131–154.

Derwent, R. G., Stevenson, D. S., Doherty, R. M., Collins, W. J., Sanderson, M. G., Johnson, C. E., Cofala, J., Mechler, R., Amann, M. and Dentener, F. J. (2005), The Contribution from Shipping Emissions to Air Quality and Acid Deposition in Europe. *Ambio, 34,* 1, 54–59.

Henesey, L., Notteboom, T. E. and Davidsson, P. (2003a), Agent-based simulation of stakeholders relations: An approach to sustainable port and terminal management. In (314-331).

Henesey, L., Wernstedt, F. and Davidsson, P. (2003b), Market-Driven Control in Container Terminal Management. In *Second International Conference on Computer Applications and Information Technology in the Maritime Industries*. Hamburg, Germany.

Itmi, M., Morel, D., Pecuchet, J.-P., Serin, F. and Villefranche, L. (1995), Eco-problem solving for containers stacking. In *Intelligent systems for the 21st century*: *Vancouver, British Columbia, Canada, October 22 - 25, 1995*. New York, NY: Inst. of Electrical and Electronics Engineers, 3810–3815.

Ji-Wen Dong and Yi-Jun Li (2003), Agent-based design and organization of intermodal freight transportation systems. In S. E. Chick (Ed.), *Proceedings of the 2003 Winter Simulation Conference*: *Fairmont Hotel, New Orleans, LA, U.S.A., December 7 - 10, 2003*. New York: Association for Computing Machinery, 2269–2274.

Kefi, M., Korbaa, O., Ghedira, K. and Yim, P. (2007), Container Handling Using Multi-agent Architecture. In *Proceedings of the 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. Wroclaw, Poland: Springer-Verlag, 685–693.

Kim, K. H. and Bae, J. W. (2004), A Look-Ahead Dispatching Method for Automated Guided Vehicles in Automated Port Container Terminals. *Transportation Science*, doi: 10.1287/trsc.1030.0082.

Kim, K. H. and Hong, G.-P. (2006), A heuristic rule for relocating blocks. *Computers & Operations Research*, doi: 10.1016/j.cor.2004.08.005.

Matthias, V., Bewersdorff, I., Aulinger, A. and Quante, M. (2010), The contribution of ship emissions to air pollution in the North Sea regions. *Environmental Pollution*, doi: 10.1016/j.envpol.2010.02.013.

Meier, L. H. (2008). *Koordination interdependenter Planungssysteme in der Logistik: Einsatz multiagentenbasierter Simulation im Planungsprozess von Container-Terminals im Hafen ; mit einem Geleitwort von Jürgen Bloech*. Wiesbaden: Gabler.

Moorthy, R. and Teo, C.-P. (2006), Berth management in container terminal: the template design problem. *OR Spectrum*, doi: 10.1007/s00291-006-0036-5.

Nishimura, E., Imai, A. and Papadimitriou, S. (2005), Yard trailer routing at a maritime container terminal. *Transport Research Part E*, 41, 53–76.

Parunak, H. V. D. (1999), Industrial and Practical Applications of DAI. In G. Weiss (Ed.), *Multiagent systems*: *A modern approach to distributed artificial intelligence*. Cambridge, Mass: MIT Press, 377–421.

Saanen, Y. A. and Dekker, R. (2006a), Intelligent stacking as way out of congested yards? Part 1. *Port Technology International*, 31, 87–92.

Saanen, Y. A. and Dekker, R. (2006b), Intelligent stacking as way out of congested yards? Part 2. *Port Technology International*, 32, 80–86.

Stuer-Lauridsen, F., Nielsen, J. B., Odgaard, T., Birkeland, M., Graugaard, C. W., Blikom, L. P., et al. (2010). *Natural gas for ship propulsion in Denmark - Possibilities for using LNG and CNG on ferry and cargo routes*. Kopenhagen, Denmark. http://www2.mst.dk/udgiv/publications/2010/978-87-92668-85-1/pdf/978-87-92668-86-8.pdf.

DEA APPLICATIONS TO PORT EFFICIENCY MEASUREMENT AND FUTURE RESEARCH POSSIBILITIES

Carlos Jahn[1] and Verena Flitsch[1]

[1]*Hamburg University of Technology, Schwarzenbergstraße 95 D, 21073 Hamburg, Germany, carlos.jahn@tu-harburg.de, verena.flitsch@tu-harburg.de*

ABSTRACT

In order to measure improvement initiatives and benchmark efficiency of ports several performance measurement tools have been developed and applied frequently. Based on a number of advantages favouring the application to a complex port system the non-parametric method Data Envelopment Analysis (DEA) evolved to be one preferential tool. The aim of this research paper is to evaluate differences of DEA applications to the port sector on the basis of 37 articles published in 21 different academic journals between 1993 and 2011 and to derive future research possibilities to measure the efficiency of ports interlinking in maritime supply chains. The main differentiation criteria proposed are (1) number and type of Decision Making Units (DMUs), (2) selected input and output criteria, (3) accessed data sources and timely data coverage, and (4) applied DEA type or combination of methods. Concluding, possible gaps in DEA applications to port supply chains are highlighted to stress future research opportunities.

Keywords: Data Envelopment Analysis (DEA), ports, efficiency measurement, future research

1 INTRODUCTION

Sea ports act as strategic units in the movement of goods in the globalised world (Ablanedo-Rosas *et al*, 2009). Ports competitiveness could affect their inherent regions' viability, prospects and propensity for growth (Munisamy, 2011), or in other words, efficiency of port operations may serve as an indicator of a country's economic development (Liu, 2008; Wu *et al*, 2010). By evaluating port efficiency, it is possible to determine its strengths and weaknesses (Liu, 2008) in order to adopt appropriate response measures in port management and operational planning. Additionally, efficiency measurement could be more than a powerful tool for port operators in constituting an input for informing regional and national port planning and operations (Al Eraqi *et al*, 2008; Cullinane *et al*, 2010).

If no engineering standard is available to define efficient performance, efficiency measurement and benchmarks are commonly applied to identify and adopt best practices as a means to improve systems performance and increase productivity (Sharma *et al*, 2010). In recent years, a method called Data Envelopment Analysis (DEA) has been increasingly applied to measure sea port efficiency (Ablanedo-Rosas *et al*, 2009). DEA is a non-parametric approach that adopts linear programming to determine an 'efficient frontier' of multiple inputs and outputs to measure the relative efficiency of Decision Making Units (DMUs, e.g. ports). The efficient frontier defines the relationship between input and output criteria by depicting the maximum output obtainable from the given inputs consumed and a DMU is considered efficient if it operates on the efficient frontier and is regarded as basically inefficient if it operates beneath the efficient frontier (Cullinane *et al*, 2005b). The main advantages of DEA according to Roll *et al (*1993) are: (1) simultaneous analysis of several inputs and outputs, (2) no requirement of a priori determination of relationships between output and inputs, (3) efficiency is measured relative to the highest performance rather than against some average, (4) highlighting of specific sub-groups of the efficient units, which are appropriate as a reference level for each of the non-efficient units. To gain further insight into the historical development and methodology of DEA please refer to Cooper *et al* (2006), or for a detailed survey and analysis of 30 years of scholarly literature in DEA until the year 2007 see also Emrouznejad *et al* (2008).

In 1993 DEA was applied to measuring the efficiency of ports for the first time by Roll *et al* and following a variety of authors revisited DEA in port research. To summarize 16 years of DEA adaption to the port sector González *et al* and Panayides *et al* published independent literature reviews in 2009. They critically analysed existing studies assessing the economic efficiency and productivity of the port sector and both identified 15 different journal articles applying DEA published between 1993 and 2006. From 2007 to 2011 publications of relevant journal articles increased significantly as more than 20 new articles came out in 5 years. Therefore, the aim of this research paper is to evaluate differences of DEA applications to the port sector on the basis of 37 articles placed in 21 different academic journals[1] between 1993 and 2011. Analysed disparities were determined as research objectives, number and type of DMUs, selected inputs and outputs, data sources and timely data coverage, and DEA type. Concluding, possible gaps in DEA applications to port supply chains are highlighted to stress future research possibilities.

2 DIFFERENCES OF DEA APPLICATIONS IN PORTS

Next to the research objective which could outline to benchmark container terminals or to evaluate the influence of security initiatives on port efficiency, the main differentiation criteria of the publications under study are: DMUs, inputs/outputs, data sources, and DEA type.

2.1 Different research objectives

The first approach of efficiency measurement with DEA in the port sector was proposed by Roll *et al* (1993) who demonstrated how relative efficiency ratings of 20 global ports can be obtained. In the following, a variety of authors concentrated on benchmarking container terminals such as Lin *et al* (2007) who applied five models of DEA to acquire a variety of complementary information about the operational efficiency of 10 major container ports in the Asia-Pacific region, and Rios (2006) who analysed the relative efficiency of 23 container terminal activities of Argentina, Brazil and Uruguay. Mixed-model approaches have been selected by Cullinane *et al* (2005b) who evaluated the efficiency of 57 entities of container ports and terminals worldwide using DEA and Free Disposal Hull model and by Sharma *et al* (2010) who analysed and demonstrated how decision tree based context-dependent DEA can be applied to 70 container terminals to enable port authorities and decision makers identify competitors, throughout deficiencies, priority variables of strategic significance to achieve and maintain competitive advantage.

Thematically, the impact of port liberalization on port efficiency has been central point in several research projects. Martinez Budria (1999) presented results of the relative efficiency of all 26 Spanish Port Authorities during the 1993-1997 years period. Diaz Hernandez *et al* (2008) measured productivity changes based on technological developments and legislative reforms in cargo handling operations in 21 Spanish ports, supplemented by Lozano *et al*

[1] Journals with more than one publication were: Maritime Economics & Logistics (9 publications), Transportation Research (3 publications), International Journal of Transport Economics (3 publications), International Journal of Shipping and Transport Logistics (3 publications), Maritime Policy and Management (2 publications), and European Journal of Operational Research (2 publications).

(2011) who determined individual and collective output targets and evaluated capital investments for 50 state-owned ports managed by 28 Spanish port authorities. Further, Barros (2003) analysed efficiency and technological change of 10 Portuguese seaport authorities. On a global level, Cullinane *et al* (2005a) presented pros and cons of port privatization and provided an empirical examination of the relationship between privatization and relative efficiency of 30 container ports worldwide. Combining different models, Estache *et al* (2004) measured the changes of efficiency since the liberalization and decentralization of public firms to regional port authorities in 11 ports of a developing country using Malmquist Total Factor Productivity Index and DEA. Bergantino *et al* (2011) verified the extent to which both governance and other non-discretionary factors affect input inefficiencies of 18 Southern European ports using a combination of DEA and Stochastic Frontier Analysis.

Next to port liberalisation, influences of other factors on port efficiency have been under study. For example, it has been of concern how available equipment (Bonilla *et al*, 2004), size and ownership (Munisamy, 2011), combination of cargo and passenger handling (Ablanedo-Rosas *et al*, 2009), security initiatives (Bichou, 2011b), global terminal operators' participation in container terminal operation (Cheon *et al*, 2010), or the internal structure of terminal operating processes for export flows (Bichou, 2011a) affect the efficiency of ports/port terminals. Directly employing determined efficiency ratings, Ferrari *et al* (2009) analysed and finally provided an alternative definition for the port concession fee through a price cap rule, which sets the fee according to the efficiency degree of each port terminal.

Broadening the scope of DEA applications from ports as single entities to the wider maritime port sector, Caillaux *et al* established in 2011 a tool to support a carrier's route choice among 110 different options of routes for containerized cargo between three given origin and destination ports, Lun *et al* (2011) examined the efficiency of liner shipping companies, and Panayides *et al* (2011) incorporated leading international maritime firms in dry, wet and container shipping.

2.2 Number and type of Decision Making Units

Two of the most distinguishing factors of DEA applications in port research are the choice and number of DMUs as summarized in table 1. Hence, the number of DMUs should be at least twice the total number of input and output variables (Lin *et al*, 2007). Exemplary, with a global perspective, Cullinane *et al* (2006) evaluated the efficiency of 25 leading container ports, and de Koster *et al* (2009) benchmarked 38 container terminals worldwide. Both papers were extended by Wu *et al* in 2010 who studied ways of retaining efficiency and improving on inefficiency of 77 container terminals in 56 ports worldwide. On a continental level, Wang *et al* (2006) investigated 104 European container terminals in 29 countries, or Hung *et al* (2010) explored efficiency estimates of 31 Asian container ports. Selecting single countries, Rios (2006) used DEA technique to analyse 23 container terminal activities of Argentina, Brazil and Uruguay. Park *et al* (2004) examined 11 Korean ports underlined by the study of Min *et al* (2008) who proposed a model to evaluate the relative efficiency of 11 Korean container terminals. Disregarding terminals and ports as DMUs, Panayides *et al* (2011) examined 26 leading international maritime firms, and Lun *et al* (2011) focused on 20 liner shipping companies.

Table 1. Number and type of DMUs

Reference	***Number and type of DMUs***
(Ablanedo-Rosas *et al*, 2009)	29 Mexican ports (18 Pacific Coast, 11 Mexican Gulf and Caribbean)
(Al Eraqi *et al*, 2008)	22 Middle East and African ports
(Barros, 2006)	24 Italian ports
(Barros *et al*, 2004)	4 Portuguese and 2 Greek ports
(Barros, 2003)	10 Portuguese port authorities
(Bergantino *et al*, 2011)	18 Southern European Ports
(Bichou, 2011a)	70 DMUs at container terminals
(Bichou, 2011b)	60 container terminals belonging to 39 ports worldwide
(Bonilla *et al*, 2004)	23 Spanish ports
(Caillaux *et al*, 2011)	3 pairs of ports and number of routes (110 possible origin-destination combinations)
(Cheon, 2009)	110 ports worldwide
(Cullinane *et al*, 2010)	25 container ports worldwide
(Cullinane *et al*, 2005a)	30 container ports worldwide
(Cullinane *et al*, 2004)	25 container ports worldwide
(Cullinane *et al*, 2005b)	57 entities, either container ports (28) or individual terminals within container ports worldwide (29)
(de Koster *et al*, 2009)	38 container terminals worldwide
(Diaz Hernandez *et al*, 2008)	21 Spanish ports
(Estache *et al*, 2004)	11 ports
(Ferrari *et al*, 2009)	8 Italian container terminals
(Hung *et al*, 2010)	31 container ports in the Asia-Pacific region
(Itoh, 2002)	8 Japanese container ports
(Lin *et al*, 2007)	10 Asia-Pacific container ports
(Liu, 2008)	10 Asia-Pacific container ports
(Lozano *et al*, 2011)	50 state-owned ports managed by 28 Spanish port authorities
(Lun *et al*, 2011)	20 liner shipping companies
(Martinez Budria, 1999)	26 Spanish port authorities in 3 sets according to port complexity (9 great, 11 medium and 6 least complexity ports)
(Munisamy, 2011)	69 Asian ports
(Min *et al*, 2008)	11 Korean container terminals
(Panayides *et al*, 2011)	26 leading international maritime firms (15 container, 6 dry bulk, and 5 tanker firms)
(Park *et al*, 2004)	11 Korean ports
(Rios, 2006)	23 container terminals (15 Brazil, 6 Argentia, 2 Uruguay)
(Roll *et al*, 1993)	20 global ports
(Sharma *et al*, 2010)	70 container terminals
(Tongzon, 2001)	4 Australian and 12 other international ports
(Turner *et al*, 2004)	22 US and 4 Canadian container ports
(Wang *et al*, 2006)	104 of Europe´s container terminals distributed across 29 countries
(Wu *et al*, 2010)	77 terminals in 56 ports worldwide (15 ports from Europe, 10 ports from North America, 6 ports of Latin America, 4 ports of Africa, 3 ports from Oceana)

2.3 Selected input and output criteria

Identification and selection of inputs and outputs is as difficult as it is crucial (Barros *et al*, 2004) because the criteria should reflect the actual objectives and processes of the DMU as accurately as possible (Cullinane *et al*, 2005a).

To begin with, the dominant output criterion in 35 of 37 articles is related to the port system's freight throughput in TEU or tons. Only Lun *et al* (2011) and Panayides *et al* (2011) act differently selecting financial figures. In contrast to outputs, input criteria are diverse. Next to technical equipment (e.g. number of cranes, straddle carriers, trucks), terminal dimensions (e.g. berth length in metres, terminal area in square metres), and labour (e.g. number of workers, working hours), financial figures (especially in mixed-model approaches) or indexes of listed input criteria were chosen. To give a comprehensive overview of differences and similarities table 2 summarizes selected inputs and outputs.

Table 2. Inputs and Outputs

Reference	***Inputs***	***Outputs***
(Ablanedo-Rosas *et al*, 2009)	Length of commercial and touristic quay line (m), labour units performing administrative and management operations, storage capacity (m²)	Number of cruise passengers, cruise ships, cargo ships, containers (TEU), volume of freight handled (tons)
(Al Eraqi *et al*, 2008)	Berth length (m), storage area (m²), handling equipment	ship calls (units), throughput (tons)
(Barros, 2006)	Personnel, capital invested (mio €), operational costs (€)	Liquid and dry bulk (tons), number of ships and passengers, containers with TEU, containers without TEU, Sales (€)
(Barros *et al*, 2004)	Labour (no. of workers), capital (book value of assets)	ships, movement of freight, total cargo handled (dry, liquid, unloaded, loaded), containers (loaded, unloaded)
(Barros, 2003)	Labour (measured by no. of workers), capital (€) (measured by the book value of assets)	Ships, movement of freight (tons), gross gauge (GT), break bulk, containerized freight, solid bulk and liquid bulk (in tons)
(Bergantino *et al*, 2011)	Quay dimension, number of terminals, area of port used for handling freight, cranes, lifters, link-belt	Port throughput (tons/year)
(Bichou, 2011a)	Szenario 1: Gate lanes, cut-off time, yard stacking index (yard stacking crane * ground storage capacity * stacking height), free yard storage, yard dwell time, STS crane index (number of quay crane * lifting capability), length overall max draft, gate and yard inputs, quay site inputs, yard dwell time Scenario 2: Gate lanes, cut-off time, gate outbound TEUs, yard stacking index, free yard storage, STS crane index, LOA max draft, gate inputs, yard and quay inputs, gate outbound TEUs	Szenario 1: Gate outbound TEUs, yard dwell time, export TEUs, STS crane move/hour, export TEUs, STS crane move/hour Szenario 2: Gate outbound TEUs, export TEUs, yard dwell time, STS crane move/hour, export TEUs, yard dwell time, STS crane move/hour
(Bichou, 2011b)	Terminal area, maximum draft, length overall, shore-to-sea quay crane index, yard stacking index, internal trucks and vehicles, number of gates	Terminal throughput (TEU)
(Bonilla *et al*, 2004)	Available equipment in million €, docks in linear metres, crane units, surface (m²), expenses in million €, no workers	Traffic in 1,000 tons
(Caillaux *et al*, 2011)	Time, costs	Options of routes
(Cheon, 2009)	Total container berth length (m), container terminal area (m²), capacity of container cranes (tonnage), aggregate hinterland size (index)	Container throughput (TEU)
(Cullinane *et al*, 2010)	Quayside gantry, yard gantry, straddle carrier	Throughput (TEU), Quay length (m), terminal area (ha)
(Cullinane *et al*, 2005a)	Terminal length (m), terminal area (ha), quayside gantry (number), yard gantry (number), straddle carrier (number)	Container throughput (TEU)
(Cullinane *et al*, 2004)	Quay length (m), terminal area (ha), quayside gantry, yard gantry, straddle carrier	Throughput (TEU)
(Cullinane *et al*, 2005b)	Quay length (m), terminal area (ha), quayside gantry, yard gantry, straddle carrier	Throughput (TEU)
(de Koster *et al*, 2009)	Number of quay gantry cranes, total quay length (m), terminal area (ha)	Annual throughput (TEU)
(Diaz	Labour (hour), crane (hour)	Containerized general cargo, non

Hernandez *et al*, 2008)		containerised general cargo, solid bulks
(Estache *et al*, 2004)	Capital (m²), Labor (workers)	Production (tons)
(Ferrari *et al*, 2009)	Total terminal surface (m²), quay length (m), sea depth (m)	Annual containers traffic (TEU)
(Hung *et al*, 2010)	Terminal area (m²), ship-shore container gantry cranes (No.), container berths (No.), terminal length (m)	Container throughput (TEU)
(Itoh, 2002)	Port infrastructure (container terminal area in m², container berths), suprastructur (gantry cranes) and labor	Import and export containers per year
(Lin *et al*, 2007)	Area of container base, no. of gantry cranes, length of container terminal number of deep-water piers	No. of vessel arrivals, loading/unloading volumes of containers
(Liu, 2008)	Container lot size, container berth length, no. of deep-water berths	Number of port calls, container cargo handled
(Lozano *et al*, 2011)	Land and stocking area, total quay length, total number of cranes, number of tugs	Total port traffic, TEUs handled, number of ship calls
(Lun *et al*, 2011)	Shipping capacity (TEU), operating costs (million US-$)	Revenue (million US-$), profit (million US-$)
(Martinez Budria, 1999)	Labor expenditures, depreciation charges, "other expenditures" = intermediate inputs and services	Total cargo moved through the docks (thousand tons), revenue obtained from the rent of facilities (millions of pesetas)
(Munisamy, 2011)	Berth length (m), terminal area (m²), total refer points, total quayside cranes, total yard equipment	Total throughput (TEU)
(Min *et al*, 2008)	Total length of quay (m), cranes, size of yard (in 1000 m²), size of labour force	Cargo throughput (in 1000 TEUs), terminal capacity (in 1000 TEUs)
(Panayides *et al*, 2011)	Sales, market value of equity, profit, book value of equity, total assets, capital expenditure, total number of employs, earnings before interest and taxes, earnings before interest, taxes, depreciation, and amortization	See inputs
(Park *et al*, 2004)	Stage 1: Berthing capacity, cargo handling capacity (million tons); Stage 2: Cargo throughput, number of ship calls; Stage 3: revenue; Stage 4: berthing capacity, cargo handling capacity	Stage 1: Cargo throughput (tons), number of ship calls; Stage 2: revenue (billion won); Stage 3: customer satisfaction (score); Stage 4: customer satisfaction
(Rios, 2006)	Number of cranes, number of berths, number of employees, terminal area	Throughput (TEU), container moves hour/ship
(Roll *et al*, 1993)	Manpower, capital, cargo uniformity	Cargo throughput, level of service, user's satisfaction, ship calls
(Sharma *et al*, 2010)	Quay length (m), terminal area (m²), quay cranes, transfer cranes, reach stacker, straddle carrier	Container throughput (TEUs)
(Tongzon, 2001)	No cranes, No berths, no tugs, terminal area, delay time, labour	Throughput, ship working rate
(Turner *et al*, 2004)	Quay length (m); terminal land (ha); container cranes (number)	TEU
(Wang *et al*, 2006)	Terminal area (ha), terminal length (m), equipment costs (million pounds)	Container throughput (TEU)
(Wu *et al*, 2010)	Capacity of cargo handling equipment, number of berths, terminal area (m²), storage capacity (TEU)	Container throughput (TEU)

2.4 Selected data sources and timely data coverage

Data was obtained from a variety of sources and can be classified into primary data (Rios, 2006; Martinez Budria, 1999), and secondary data e.g. from statistical reports of national authorities (Turner *et al*, 2004; Park *et al*, 2004; Estache *et al*, 2004; Barros, 2003; Barros, 2006; Barros *et al*, 2004), or secondary data obtained in 17 of 37 articles from

‘Containerisation International Yearbook’ (Bergantino *et al*, 2011; Bichou, 2011b; Hung *et al*, 2010; Lin *et al*, 2007; Liu, 2008; Lun *et al*, 2011; Munisamy, 2011; Sharma *et al*, 2010; Tongzon, 2001; Turner *et al*, 2004; Wang *et al*, 2006; Wu *et al*, 2010; Cheon, 2009; Cullinane *et al*, 2010; Cullinane *et al*, 2005a; Cullinane *et al*, 2004; Cullinane *et al*, 2005b).

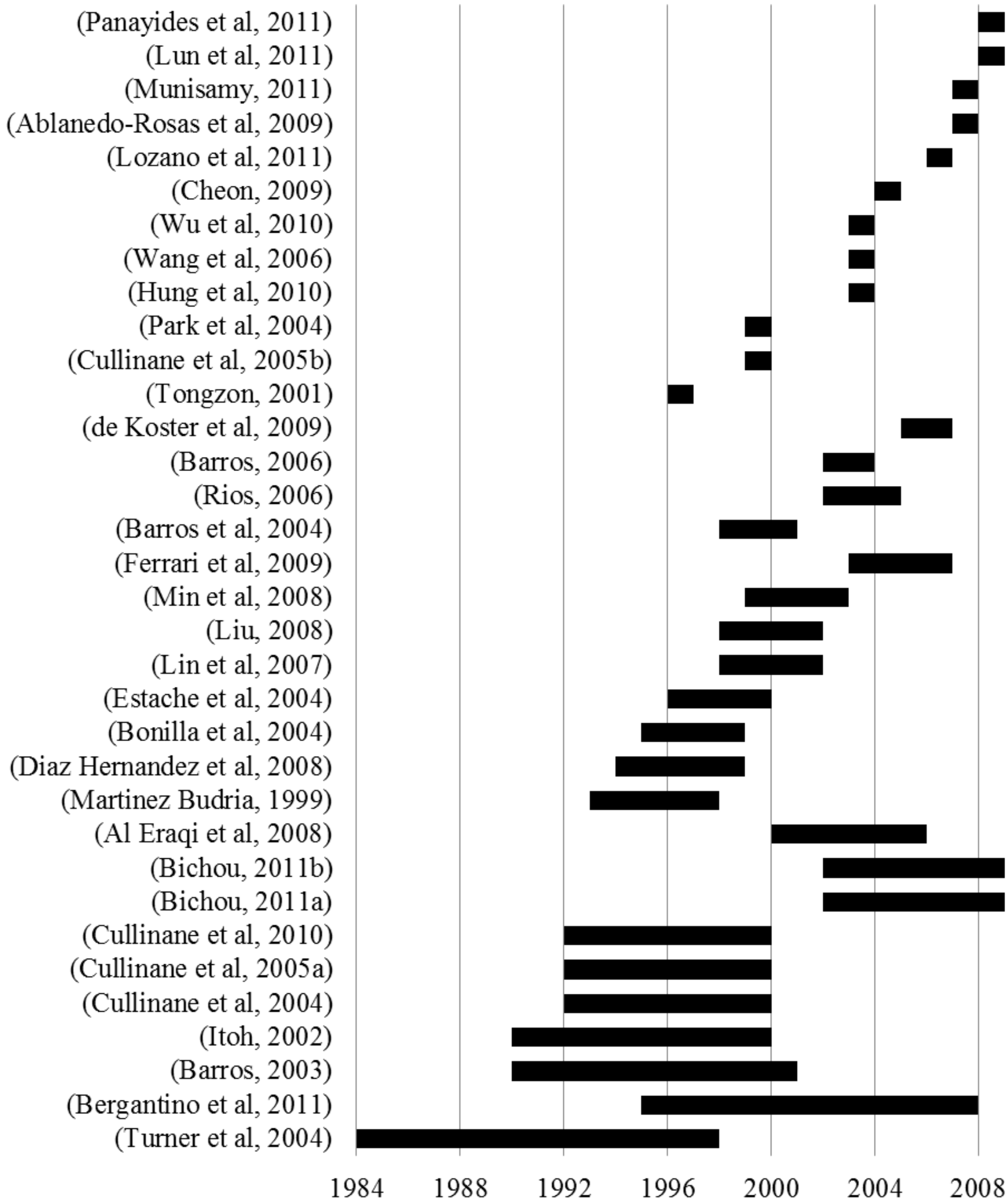


Figure 1. Time horizon of data in DEA applications

The time horizon of data varies from cross-sectional data to time-series and is displayed in figure 1. 12 out of 34 publications[2] selected cross-sectional data from one year (35 %), two publications choose two years (6 %), two three years (6 %). 14 publications undertook time-series research with data from four up to eight years (41 %), and four publications selected data for more than 10 years (12 %). Bergantino *et al* (2011) and Turner *et al* (2004) evaluated

[2] Due to fictional data or incomprehensive information about the time frame Roll *et al* (1993), Caillaux *et al* (2011), and Sharma *et al* (2010) were excluded.

efficiency of container ports applying the longest time-series data for an 11- and a 14-years period respectively.

2.5 Applied DEA type

Generally, DEA models can be distinguished into the CCR and the BCC model[3]. DEA-CCR is based on the consumption of constant returns to scale so that all observed production combinations can be scaled up or down proportionally (Cullinane *et al*, 2010). DEA-BCC is constructed on the concept of variable returns to scale and is geographically represented by a piecewise linear convex frontier (Cullinane *et al*, 2010).

The CCR model was adopted by five analysed publications (Caillaux *et al*, 2011; Lun *et al*, 2011; Wu *et al*, 2010; Ablanedo-Rosas *et al*, 2009; Roll *et al*, 1993). BCC-model was also chosen by five authors (Cullinane *et al*, 2010; Ferrari *et al*, 2009; Rios, 2006; Bonilla *et al*, 2004; Martinez Budria, 1999). Following a mixed-model approach, combinations of CCR and BCC models were utilised in 15 articles (Al Eraqi *et al*, 2008; Barros *et al*, 2004; Bichou, 2011a; Cullinane *et al*, 2005a; Cullinane *et al*, 2004; de Koster *et al*, 2009; Liu, 2008; Tongzon, 2001; Barros, 2006; Lin *et al*, 2007; Munisamy, 2011; Wang *et al*, 2006; Itoh, 2002; Min *et al*, 2008; Park *et al*, 2004).

Integrating other methods than DEA, 10 researchers selected mixed-method approaches inheriting Stochastic Frontier Analysis (Bergantino *et al*, 2011; Panayides *et al*, 2011), Free Disposable Hull Analysis (Cullinane *et al*, 2005b), Total Frontier Productivity Index (Diaz Hernandez *et al*, 2008), Bootstrap algorithm (Hung *et al*, 2010), decision-trees (Sharma *et al*, 2010), Tobit regression (Turner *et al*, 2004), and Malmquist Productivity Index (Bichou, 2011b; Estache *et al*, 2004; Barros, 2003). No clear classification could be determined for Cheon (2009) and Lozano *et al* (2011).

3 RESEARCH GAPS

In regard to gaps in DEA applications to the port sector, attention could be drawn to possible research aims and DEA set ups. As efficiency ratings of container terminals in the Asia-Pacific region and on a global level have been performed several times, Barros (2006) can be supported who argued that benchmarking the European seaport sector with DEA is a relatively under-researched area. Especially the North Range, Baltic, German or Dutch ports have not been in the centre of academic debate. In the long-run, benchmarking efficiencies of European ports including the new container terminal in Wilhelmshaven (scheduled opening: August 2012) is of value.

Most commonly, port liberalization has been analysed from a port authorities' perspective, but developments with respect to efficiency measurement of ocean carriers' involvement in dedicated terminal structures demand further investigation beyond the research of Cheon *et al* (2010). Similar to Ablanedo-Rosas *et al* (2009) who emphasised the influence of combined cargo and passenger traffic on ports, the impact of growing cruise shipping volumes for cargo handling facilities or port destinations could be assessed. Moreover, the effect of security initiatives was evaluated by Bichou, (2011b), but the aftermaths of environmental standards on ports such as the 'International Convention for the Prevention of Pollution from Ships (MARPOL)', which exemplary enables cold ironing and ballast water treatment, are still unobserved.

Many ports have evolved to become pivotal nodes in international logistics networks and product supply chains (Wang *et al*, 2006). To achieve best-practice performance, managing terminals as integrated operating sites could be essential (Bichou, 2011a). Accordingly, efficiency measurement of port integration in supply chains offers research potential

[3] BCC and CCR are abbreviations of the names 'Banker, Charnes, Cooper' and 'Charnes, Cooper, Rhodes'.

following up the internal structure analysis of terminal operating processes for export flows from Bichou (2011a). Ports act as service providers, in particular for vessels, cargo and inland ports, therefore, port performance cannot be assessed on the basis of a single value or measure (Cullinane *et al*, 2004). Thus, benchmarking multimodal services, exemplary efficiency of truck, rail or barge processing, or establishing tools to support a carrier's choice among different port locations in analogy to Caillaux *et al* (2011) are of importance. Examining efficiency of sea freight hauliers following Lun *et al* (2011) and Panayides *et al* (2011) could highlight another performance perspective of the maritime transport chain.

Based on illustrated possible research objectives, determination of DMUs would extent from container terminal selection to choice of other maritime supply chain stakeholders. One limitation of DEA is that efficiency values are relative not absolute. This necessitates completeness of data gathered (Liu, 2008). Selecting primary data with long-time coverage next to statistical series could help to overcome this vulnerability. In addition, combinations of DEA with other techniques to measure efficiency such as Stochastic Frontier Analysis, Total Frontier Productivity Index, or Malmquist Productivity Index increase validity and support overcoming different methodological drawbacks, as well as combining advantages.

4 CONCLUSION

Efficiency measurement of ports with DEA can enable adoption of best practices as a means to improve systems performance and increase productivity in a competitive environment. DEA has been applied frequently between 1993 and 2011 with different research objectives reaching from measuring container terminal efficiency on a global basis to benchmarking origin-destination combinations of ports in regional areas. Differences in selection of DMUs and inputs/outputs either distinguish studies with similar research aims like efficiency measurement of container terminals, or they clearly separate research approaches according to the benchmarking of different entities such as maritime companies, liner shippers, or export flows. Data sources are mainly based on statistical data from 'Containerisation International' or other statistical sources and traditional DEA types or combinations with other methods were chosen.

In summary, the identification of gaps in DEA applications to the port sector promotes further research which concentrates the attention to other geographical areas or to alternative business sectors following a supply chain perspective with ports as integrated nodes in global transport networks. To overcome drawbacks of DEA and generate synergies, long-time studies and mixed-method approaches are recommended.

REFERENCES

Ablanedo-Rosas, J. H. and Ruiz-Torres, A. J. (2009), Benchmarking of Mexican ports with data envelopment analysis, *International Journal of Shipping and Transport Logistics*, **I 1**, 276–294.

Al Eraqi, A. S., Mustafa, A., Khader, A. T. and Barros, C. P. (2008), Efficiency of middle eastern and east African seaports: application of DEA using window analysis, *European Journal of Scientific Research*, **I 23**, 597–612.

Barros, C. P. (2003), The measurement of efficiency of portuguese sea port authorities with DEA, *International journal of transport economics*, **I 30**, 335–354.

Barros, C. P. (2006), A benchmark analysis of Italian seaports using data envelopment analysis, *Maritime Economics & Logistics*, **I 8**, 347–365.

Barros, C. P. and Athanassiou, M. (2004), Efficiency in European seaports with DEA: evidence from Greece and Portugal, *Maritime Economics & Logistics*, **I 6**, 122–140.

Bergantino, A. S. and Musso, E. (2011), The role of external factors versus managerial ability in determining seaports' relative efficiency: An input-by-input analysis through a multi-step approach on a panel of Southern European ports, *Maritime Economics & Logistics*, **I 13**, 121–141.

Bichou, K. (2011a), A two-stage supply chain DEA model for measuring container-terminal efficiency, *International Journal of Shipping and Transport Logistics*, **I 3**, 6–26.

Bichou, K. (2011b), Assessing the impact of procedural security on container port efficiency, *Maritime Economics & Logistics*, **I 13**, 1–28.

Bonilla, M., Casasus, T., Medal, A. and Sala, R. (2004), An efficiency analysis with tolerance of the Spanish port system, *International journal of transport economics*, **I 31**, 379–400.

Caillaux, M. A., Sant'Anna, A. P., Meza, L. A. and Soares Mello, J. C. C. B. de (2011), Container logistics in Mercosur: Choice of a transhipment port using the ordinal Copeland method, data envelopment analysis and probabilistic composition, *Maritime Economics & Logistics*, **I 13**, 355–371.

Cheon, S. (2009), Impact of global terminal operators on port efficiency: A tiered data envelopment analysis approach, *International journal of logistics*, **I 12**, 85–101.

Cheon, S., Dowall, D. E. and Song, D.-W. (2010), Evaluating impacts of institutional reforms on port efficiency changes ownership, corporate structure, and total factor productivity changes of world container ports, *Transportation research*, **I 46**, 546–561.

Cooper, W. W., L. M. Seiford and K. Tone (2006). *Introduction to data envelopment analysis and its uses: With DEA-solver software and references*. New York, NY.

Cullinane, K. and Teng-Fei Wang (2006), The efficiency of European container terminals and implication for supply chain management, *Maritime Economics & Logistics*, **I 8**, 82–99.

Cullinane, K. and Wang, T. (2010), The efficiency analysis of container port production using DEA panel data approaches, *OR Spectrum*, **I 32**, 717–738.

Cullinane, K., Ji, P. and Wang, T.-F. (2005a), The relationship between privatization and DEA estimates of efficiency in the container port industry, *Journal of economics and business*, **I 57**, 433–462.

Cullinane, K., Song, D.-W. and Wang, T. (2005b), The application of mathematical programming approaches to estimating container port production efficiency, *Journal of productivity analysis*, **I 24**, 73–92.

Cullinane, K., Song, D.-W., Ji, P. and Wang, T.-F. (2004), An Application of DEA Windows Analysis to Container Port Production Efficiency, *Review of Network Economics*, 184–206.

de Koster, M. B. M., Balk, B. M. and van Nus, W. T. I. (2009), On using DEA for benchmarking container terminals, *International journal of operations & production management*, **I 29**, 1140.

Diaz Hernandez, J., Martinez-Budria, E. and Jara-Diaz, S. (2008), Productivity in cargo handling in Spanish ports during a period of regulatory reforms, *Networks and spatial economics*, **I 8**, 287–295.

Emrouznejad, A., Parker, B. R. and Tavares, G. (2008), Evaluation of research in efficiency and productivity: A survey and analysis of the first 30 years of scholarly literature in DEA, *Socio-Economic Planning Sciences*, **I 42**, 151–157.

Estache, A., Tovar, B. de and Trujillo, L. (2004), Sources of efficiency gains in port reform: a DEA decomposition of a Malmquist TFP index for Mexico, *Utilities Policy*, **I 12**.

Ferrari, C. and Basta, M. (2009), Port concession fees based on the price-cap regulation: a DEA approach, *Maritime Economics & Logistics*, **I 11**, 121–135.

González, M. M. and Trujillo, L. (2009), Efficiency measurement in the port industry: a survey of the empirical evidence, *Journal of transport economics and policy*, **I 43**, 157–192.

Hung, S.-w., Lu, W.-m. and Wang, T.-p. (2010), Benchmarking the operating efficiency of Asia container ports, *European journal of operational research*, **I 203**, 706–713.

Itoh, H. (2002), Effeciency Changes at Major Container Ports in Japan: A Window Application of Data Envelopment Analysis, *RURDS. Review of urban and regional development studies*, **I 14**, 133–152.

Lin, L. C. and Tseng, C. C. (2007), Operational performance evaluation of major container ports in the Asia-Pacific region, *Maritime policy and management*, **I 34**, 535–551.

Liu, C. C. (2008), Evaluating the operational efficiency of major ports in the Asia-Pacific region using data envelopment analysis, *Applied economics*, **I 40**, 1737.

Lozano, S., Villa, G. and Canca, D. (2011), Application of centralised DEA approach to capital budgeting in Spanish ports, *Computers & industrial engineering*, **I 60**, 455–466.

Lun, V. Y. H. and Marlow, P. (2011), The impact of capacity on firm performance: A study of the liner shipping industry, *International Journal of Shipping and Transport Logistics*, **I 3**, 57–71.

Martinez Budria, E. (1999), A study of the efficiency of Spanish port authorities using data envelopment analysis, *International journal of transport economics*, **I 26**.

Min, H. and Park, B.-i. (2008), A hybrid Data Envelopment Analysis and simulation methodology for measuring capacity utilisation and throughput efficiency of container terminals, *International journal of logistics systems and management*, **I 4**, 650–672.

Munisamy, S. (2011), Benchmarking the efficiency of Asian container ports, *African journal of business management*, **I 5**, 1397–1407.

Panayides, P. M., Lambertides, N. and Savva, C. S. (2011), The relative efficiency of shipping companies, *Transportation research*, **I 47**, 681–694.

Panayides, P. M., Maxoulis, C. N., Wang, T.-F. and Ng, K. Y. A. (2009), A critical analysis of DEA applications to seaport economic efficiency measurement, *Transport reviews*, **I 29**, 183–206.

Park, R. K. and De, P. (2004), An alternative approach to efficiency measurement of seaports, *International Journal of Maritime Economics*, **I 6**, 53–69.

Rios, L. R. (2006), Analysing the relative efficiency of container terminals of Mercosur using DEA, *Maritime Economics & Logistics*, **I 8**, 33346.

Roll, Y. and Hayuth, Y. (1993), Port performance comparison applying data envelopment analysis (DEA), *Maritime policy and management*, **I 20**, 153–161.

Sharma, M. J. and Yu, S. J. (2010), Benchmark optimization and attribute identification for improvement of container terminals, *European Journal of Operational Research*, **I 201**, 568–580.

Tongzon, J. (2001), Efficiency measurement of selected Australian and other international ports using data envelopment analysis, *Transportation Research Part A*, **I 35**, 107–122.

Turner, H., Windle, R. and Dresner, M. (2004), North American containerport productivity: 1984-1997: Shipping Finance and Port Issues, *Transportation Research Part E: Logistics and Transportation Review*, **I 40**, 339–356.

Wang, T.-F. and Cullinane, K. (2006), The Efficiency of European Container Terminals and Implications for Supply Chain Management, *Maritime Economics & Logistics*, **I 8**, 82–99.

Wu, J., Yan, H. and Liu, J. L. (2010), DEA models for identifying sensitive performance measures in container port evaluation, *Maritime Economics & Logistics*, **I 12**, 215–236.

TRANSPORT SYSTEMS FOR LINKING SEAPORT CONTAINER TERMINALS AND DEDICATED SATELLITE TERMINALS

Joachim R. Daduna[1,†], Robert Stahlbock[2] and Stefan Voß[3]

[1]*Berlin School of Economics and Law, Badensche Str. 52, D-10825 Berlin, Germany, daduna@hwr-berlin.de*

[2]*University of Hamburg, Inst. of Information Systems, Von-Melle-Park 5, D-20146 Hamburg, Germany; also Lecturer at FOM University of Applied Sciences, Essen / Hamburg, Germany, stahlbock@econ.uni-hamburg.de*

[3]*University of Hamburg, Inst. of Information Systems, Von-Melle-Park 5, D-20146 Hamburg, Germany, stefan.voss@uni-hamburg.de*

ABSTRACT

For many seaports considerable problems arise when it comes to expanding their facilities as the needed areas are not available due to historically grown urban settlement structures and their (usually more profit-oriented) forms of land use. These circumstances limit the quantitative and qualitative performance enhancement, which is, nevertheless, fundamental for international competition in times of increased demand. Hereof *seaport container terminals* (SCT) are strongly affected because process structures in container transport require sufficient space and fitting technological equipment. A repeatedly discussed solution persists in building *satellite terminals* (ST), which are established in the close vicinity of the SCT. Nevertheless, one critical aspect of this solution is the design of a sufficient transport infrastructure between SCT and ST, which, on one hand, must be enforceable from the transport policy point of view and, on the other hand, must be able to secure a sufficient performance.

Here (fully automated) railway and (guided) road traffic systems, heavy-lift cable car and overhead conveying systems as well as underground tunnel systems are under consideration for container transport. Existing as well as conceptualized systems are analyzed and assessed with regards to their possible usability for shuttle transports between SCT and ST. Fundamental for this are operational aspects (i.e. accessible capacity, velocity and security), the question of technical feasibility, arising investment and operational costs as well as (political) enforceability. Based on these aspects it is determined which of the possible solutions is the most qualified one to link the SCT in the port of Hamburg and potentially to-be-established ST.

Keywords: container terminals, transport systems, hinterland connection, satellite terminals

1 DEVELOPMENT TENDENCIES IN SEAPORT TERMINALS

Due to the beginning worldwide financial crisis in 2008 the long-lasting and continuous increase of international container traffic showing a partly extreme boost of throughput within SCT (see, e.g., UNCTAD 2009, 2010, 2011) has stopped. Meanwhile the decrease in container transport in 2009 has been widely compensated or even already over-compensated. Nevertheless, it is no longer estimated to achieve growth rates larger than 200% as projected for 2004-2025 for the port of Hamburg a few years ago (see, e.g., PLANCO Consulting 2007:70). Nevertheless, most of the *seaport container terminals* (SCT) are still threatened by capacity bottlenecks, which are already emerging or which can be expected in the next years.

---

† Corresponding author: e-mail : daduna@hwr-berlin.de

As many SCT cannot be expanded to the needed extent due to the given (often historically grown) *urban settlement structures*, new solutions have to be sought. Two approaches under discussion are on one hand the installation of *satellite terminals* (ST) in the close vicinity of an SCT (see, e.g., Daduna 2011) and on the other hand the usage of given transshipment terminals in the hinterland (see, e.g., Roso *et al.* 2009, and the related discussion on dry ports). Here, the essential target is a reduction of temporary storage within the SCT by relocating handling activities into the hinterland (see, e.g., Schwartz *et al.* 2009). The discussions about integrating ST in order to disburden the SCT in the port of Hamburg are the initial point of the subsequent investigation. At first the exploitability of ST is given a closer examination, whereupon the essential question is how to link and integrate traffic with the SCT. For this, different shuttle systems are then introduced and analyzed. In the following steps these systems are discussed with regards to their application using the example of Hamburg's port.

2 SATELLITE TERMINALS AS CAPACITY EXPANSION

The central idea of the ST-concept is based on an area-related dislocation of container yards in connection with a relocation of handling activities (see, e.g., Slack 1999, Dimitrijevic and Spasovic 2006, Roso 2008, Franke 2008, Rosa and Roscelli 2009, Daduna 2011). Depending on an SCT's degree of utilization, for instance, incoming import containers will be unloaded and immediately transported into an ST via a shuttle connection. Here, these containers can be stored intermediately until further transport in a way they would be stored in an SCT. In this case the ST has the function of a yard extension. With an appropriate process organization an ST can also be operated as cross docking facility as long as the hinterland transport is conducted via rail freight transport. In this case the containers will be directly moved from the shuttle system onto available block trains, which operate the different hinterland terminals at individual service. Fig. 1 shows the basic process structure while using an ST for handling import containers.

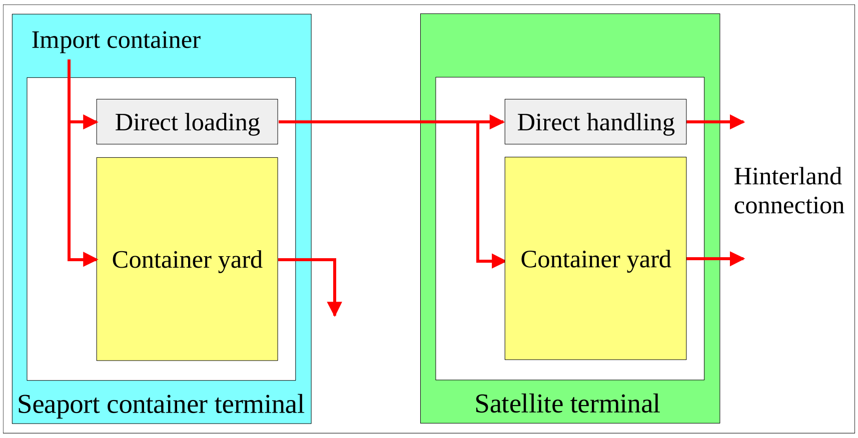


Figure 1. Operating import containers

In particular the cross docking option has considerable advantages in operations but also high requirements in terms of designing multi-modal transportation processes. If it is possible to synchronize the operations in a convenient way, shipping times of containers can be shortened extensively by continuous transport and the elimination of interim storage. Apart from that the need of storage capacities at the ST is reduced.

While handling export containers the ST can function as a buffer, i.e. the containers for the SCT can be stored intermediately at the ST and be forwarded on demand. By operating the shuttle system, the SCT is served just-in-sequence; this means the containers are sent directly to the quay and are provided simultaneously on time for loading. Under certain circumstances a direct (and punctual) supply from the hinterland can be carried out, but this requires an enormous amount in planning, monitoring, and control. Fig. 2 shows the basic structure of these processes.

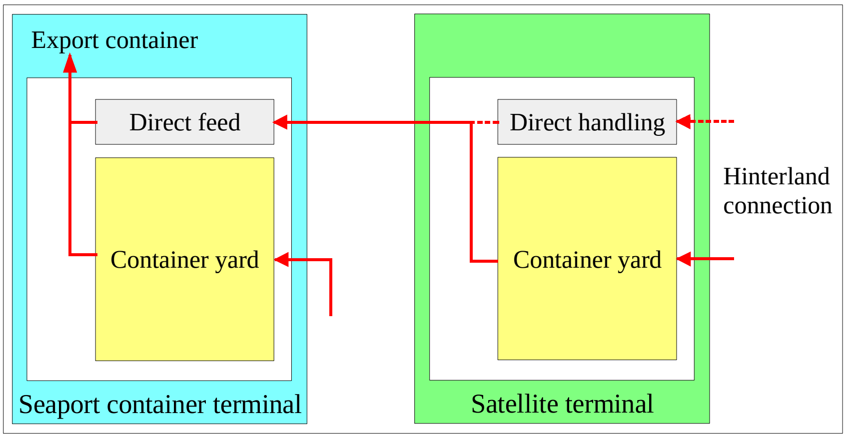


Figure 2. Operating export containers

The outlined structures and fields of application illustrate the potentials that result from incorporating ST into container flow design. The central point of the implementation of such a concept is the performance of the applied shuttle system as well as its investment and operational costs. Hence, in the following section the different alternatives are introduced and analyzed.

3 DESIGN OF SHUTTLE CONNECTIONS

Several solutions exist for the implementation of shuttle connections which are based upon varying technical concepts. Here, essential aspects are the spatial structure of the track system and the drive concept as well as the functionality and performance. Overall, there are six alternate transport systems presented below. Nevertheless, not all of them have been realized so far or they were not intended for the transport of loaded containers, respectively.

- **Alternative 1: Shuttle trains**

  This alternative describes a point-to-point service with block trains between SCT and ST. As a rule, one can assume in this case an existing network and transshipment infrastructure as well as given train systems. Nonetheless, it is legally prohibited to automate the transportation processes if a conventional rail transport is conducted simultaneously on the same rail tracks. Apart from that qualification measures for increasing the capacities and the purchase of additional shuttle trains in order to cover the rising transport demand could become necessary. In this concept the connection between quay and rail head in the SCT might become a problem if the rail tracks do not directly lead onto the quays, because the currently used vehicles and transshipment facilities are not configured for coping with the underlying demand peaks, which could appear while using shuttle transfer. A possible way of solving this problem could be to install a heavy-lift overhead-conveying system (see alternative 5), which is not only a transportation system but also a transshipment system.

  In addition, the application of shuttle trains constitutes an excellent basis for a cross docking solution, which is predicated on a MegaHub concept (see, e.g., Alicke 2002; Bontekoning 2006: pp. 40; Franke 2008). Furthermore, it is likely to accomplish a fundamental shift from hinterland container transport to rail, in particular if a Europe-wide network of high-performance facilities for rail / rail-transshipments will be installed in the future (see, e.g., Rodrigue 2008, Limbourg and Jourquin 2009).

- **Alternative 2: Automated guided vehicles on separate tracks**

  The necessary technical standard for the implementation of *automated guided vehicles* (AGV) is available and already under discussion for several projects, for instance the *CargoRail* concepts (see, e.g., Dimitrijevic and Spasovic 2006), the *Electric Cargo Conveyor* (ECCO) (see, e.g., James 2008) (see Fig. 3) and the *SAFE Freight Shuttle* (see, e.g., Roop *et al.* 2011) (see Fig. 4). One can distinguish between an operation using electric powered solo vehicles or trains (see, e.g., Schönemann and Plattner 2012). The latter can be either coupled single vehicles and / or tractors with trailers. This solution, nonetheless, requires the installment of a system-specific network infrastructure, as a simultaneous use of the public road infrastructure with manually and automatically driven vehicles is legally prohibited. A transport service via (conventionally or electrically motorized) trucks does not make sense for economic reasons (e.g., high personnel costs). Apart from that the impact of traffic on the infrastructure would increase immensely between SCT and ST; from a transport policy and environmental perspective this is usually not desirable.

Figure 3. Electric Cargo Conveyor (ECCO) (model)
Source: http://atg.ga.com/EM/transportation/images/main1.jpg

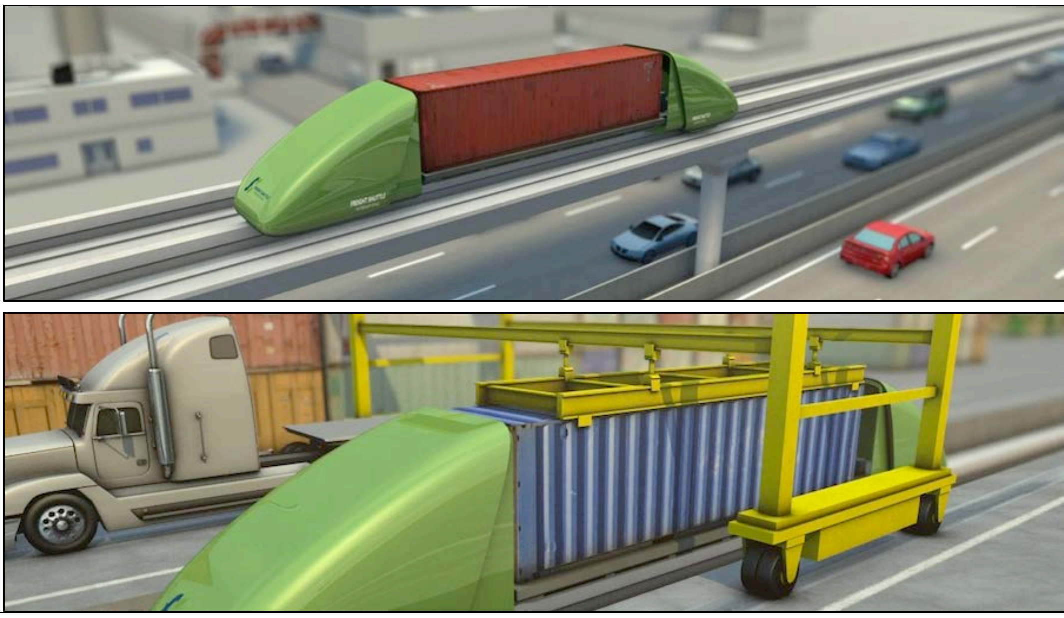


Figure 4. SAFE Freight Shuttle (model)
Sources: http://www.freightshuttle.com/benefits/; http://www.freightshuttle.com/concept/

- **Alternative 3: Cable car systems**

  The drive technology of these systems, which are usually employed for passenger transport is implemented into the infrastructure as it consists of a locally fixed cable-based system. Hence, the vehicles do not have an independent driving system and, therefore, they have a lower weight. In such a system the investment costs for the track infrastructure increase while the costs for the single vehicles decrease, i.e., the costs are shifted within the system. However, an application for container transportation within an SCT is also under discussion (see, e.g., Pallasch *et al.* 2010), but not yet put into operation.

- **Alternative 4: Heavy-lift aerial ropeway systems**

  This transport system is mainly used for passenger transport while it is currently deployed for freight transport in a very limited way. In the foreground of this technique are implementations for in-plant transport operations within large-scale industrial plants with high (and usually continuous) material flows (see, e.g., Pallasch *et al.* 2010, Schönemann and Plattner 2011). Another field of implementation exists for areas with a difficult topography and insufficient access, e.g. in mountain areas for servicing (often temporarily occurring) freight transport measures in 3-dimensional space. As in alternative 5, the land consumption stays comparably small.

- **Alternative 5: Heavy-lift overhead-conveying systems**

  This alternative is about the (not yet realized) concepts Concar (see, e.g., Arnold and Rall 1996) (see Fig. 5) as well as Auto-GO (see, e.g., Dimitrijevic and Spasovic 2006) (see Fig. 6), which are based on common (floor-free) electric overhead-conveyers for in-plant transportation. One of their main advantages is the combination of transshipment and transportation in a single system that is, for instance, notably qualified as an interface to rail freight transport. In this respect it is possible to load and unload the trains used in hinterland transport in just one step within the ST, if the corresponding system capacity is given. However, this system-inherent advantage is lost if intermediate storage of containers is required (even if it is required for a short time only). Other advantages are the lower land use and, in addition, a reduced land sealing (compared to alternatives 2 and 3) due to the elevated track system.

Figure 5. **(**Heavy-lift) Overhead-conveying system Concar (model)
Source: Arnold and Rall 1996

Figure 6. (Heavy-lift) Overhead-conveying system Auto-GO (model)
Source: Dimitrijevic and Spasovic 2006

- **Alternative 6: Underground transportation systems**

  A number of technical approaches exists for applying underground freight transport systems (see, e.g., Pielage and Rijsenbrij 2005, Dellmann and Berger 2006, Visser *et al.* 2008), but none of them have been implemented for practical operations. Concepts for harbor-integrated container transports (see, e.g., Braet 2011) are, until now, also only under discussion. Basically, these tunnel systems have some advantages. There are no conflicts given between tunnel systems and above-ground traffic infrastructure or existing urban settlement structures, e.g. with respect to possible noise disturbance. As the system is closed (to a large extent), there are other advantages. At first, the operations are independent from weather influences so that unexpected operational disturbances are not very likely. Moreover, this solution constitutes a high protection against external influences of third parties that could occur in other circumstances. An example of an underground container transport concept is shown in Fig. 7.

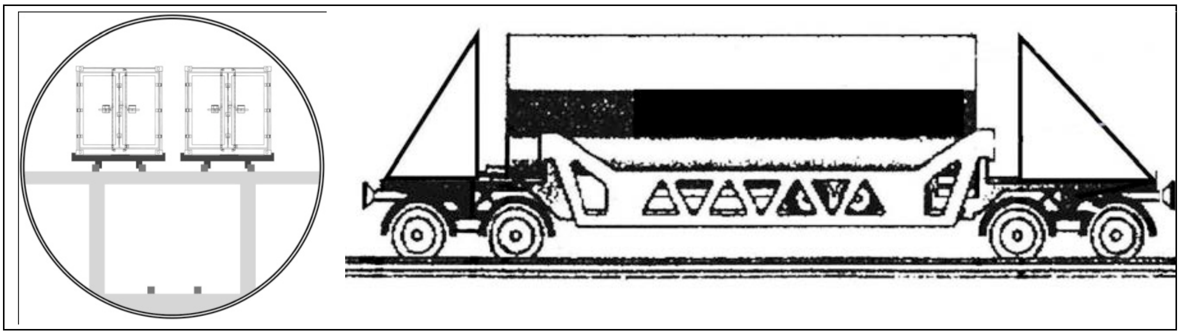


Figure 7. Underground container transport concept (model; cross-section of tunnel and longitudinal cut of transport vehicle)
Source: Marnette and Böwer 2011 – modified

All six alternatives are fundamentally possible solutions for implementing shuttle systems for container transportation; nevertheless all of them have to be evaluated by technical, operational and economical aspects. Additionally, the question of (political) enforceability concerning necessary traffic infrastructure measures arises, as these are usually affiliated with antagonisms from different sides. Apart from that the time frame for planning, approval procedures and constructional realization has to be considered, as in the most important SCT enormous capacity problems have already occurred or at least are expected to arise in the medium term.

4 SHUTTLE CONNECTION FOR THE SEAPORT CONTAINER TERMINALS OF HAMBURG

It is expected that the existing SCT *Altenwerder* (CTA), *Burchardkai* (CTB), *Eurogate* (CTE) and *Tollerort* (CTT) will be faced with capacity problems (for a throughput of about nine million TEU in the year 2011), as a sufficient extension of the facilities seems not possible. Even the additionally planned terminal *Steinwerder* (CTS) will not be an adequate solution for this problem. This is why suitable measures have to be seized, especially with respect to increasing amounts of throughput in the following years. On this account the involvement of ST is also discussed in Hamburg, where the shunting yard *Maschen* (STM) and the *Logpark Hamburg* in the area of *Rade* (STR) seem to be the two best-suited facilities. Fig. 8 depicts the regional structures.

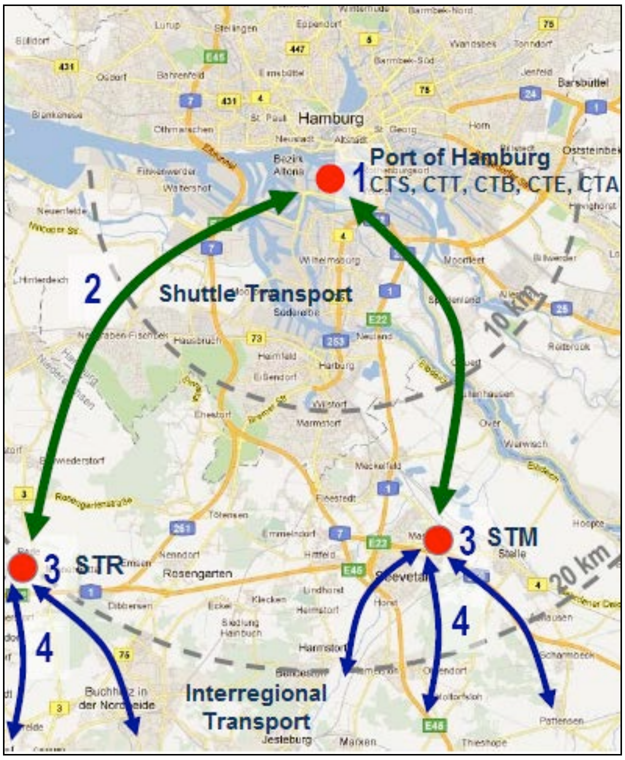


Figure 8. Regional structure of the seaport and the hinterland
Sources: https://maps.google.com; Marnette and Böwer 2011 – modified

A currently expected transport demand of 5.0 to 10.0 million TEU per year (calculated for 7,000 operating hours) results in an amount of 720 to 1,440 TEU per hour for the shuttle connection. For the track line section between the connecting node of the main line and the link from (or to) the CTA and the branching node (or rather the merging node) to the hinterland facilities STM and STR (Fig. 8), the transport system has to be constructed for a headway of five seconds. Assumed is a vehicle dimensioning of two TEU with a maximal load capacity of up to 50 tons. The required fleet size for covering the demand (including additional standby vehicles) depends on the (technical) operating speed, the distance and the duration of transshipments. Fig. 9 shows a basic concept for the required track line connecting the SCT area with both discussed STs, whereas one can assume a total length of 75 kilometers (for double-track operation).

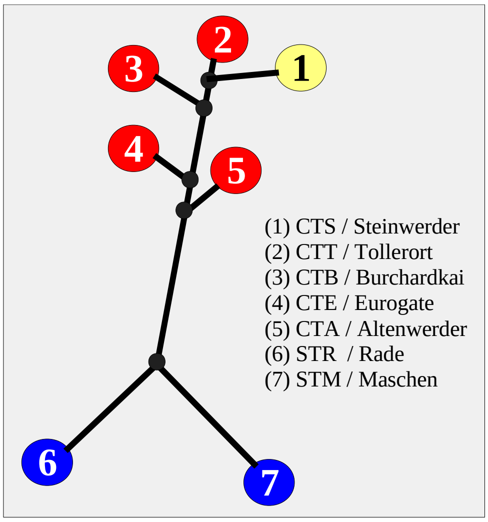


Figure 9. Basic structure of a track line concept (for implementing two ST)

5 ANALYSIS AND EVALUATION

For analyzing and evaluating the alternatives, the focus is on the basic feasibility, considering technical and operational aspects. But in this context only very rough estimations can be given for investment and operational costs. An authentic approximation is impossible before final decisions are made about the alignment and configuration of the track system as well as the technical design of the vehicles.

- **Evaluation of the application of shuttle trains (Alternative 1)**

  The maximal dimensioning of 1,440 TEU/h results in a demand of 14 (incoming and outgoing) shuttle trains per hour, assuming a train length of about 700 meters with 26 two-part container flat wagons (type: Sggrss 734**)** and a capacity of 104 TEU per train. It would be necessary to provide (or partly purchase) a corresponding number of rolling stock. While this would be realizable, the major remaining problem is the insufficient network and transshipment capacity.

  The track line southerly from the seaport area, for instance, is already heavily strained, primarily through local and long-distance passenger rail transport (with increasing tendencies). As for track access freight transport has a hierarchically lower priority than passenger transport, the required track capacities are often not available, in particular during traffic peaks. A corresponding development of the track system is under the current circumstances neither financially feasible nor politically enforceable without even giving regard to the question of the potential time frame. Apart from that an extensive capacity increase within the SCT would be indispensable. The second set of critical points is the in-plant activities within the SCT, namely the transport of import containers to the transshipment facility as well as the available technical systems and their performance level. Innovative transport and transshipment techniques have to be developed and implemented in order to solve these problems. The concept of Concar (see above) constitutes a suitable basis.

- **Evaluation of the application of automated guided vehicles on separate tracks (Alternative 2)**

  In order to implement such an AGV concept it is necessary to build up a (totally) new infrastructure, which leads in comparison to alternative 6 to lower investment costs because of the elevated construction with pre-assembled segments. As this concept includes a direct connection of the transport system to the quay area, the handling effort decreases considerably resulting in additional advantages with respect to the transshipment procedure. The required performance will be provided by deploying AGV with electrical traction and a load capacity of 50 tons maximum and a length of 14 meters (for containers up to 45'). The implementation of such a transport system is technically feasible with regards to infrastructure as well as required vehicles.

Given the maximal transport performance of 1,440 TEU/h the headway on the most heavily used track section amounts to five seconds, provided that the available vehicle capacity of two TEU is fully used on every tour. Even a smaller space is possible by establishing an appropriate control and signaling system. Nevertheless, constraints can occur if the transshipment performance at the quay area is not sufficiently dimensioned or the possible number of simultaneously running transshipments is too small to accomplish the required throughput.

- **Evaluating the application of cable car systems (Alternative 3)**

  These systems are basically proper for heavy-lift transports for load capacities up to 50 tons as well as for dense headways of single cable cars. Due to technical restrictions, as for example the driving power and the resultant maximal length of a track section as well as its accelerating ability, the required performance as described in Section 4 cannot be attained.

- **Evaluation of the application of heavy-lift aerial ropeway systems (Alternative 4)**

  The range of performance can be compared to the one of alternative 3, i.e., these systems cannot fulfill the demands for the needed transport capacity.

- **Evaluation of the application of heavy-lift overhead-conveying systems (Alternative 5)**

  This system which is based on a comparatively simple method regarding infrastructure and vehicle system can be implemented. Given a correspondingly constructed track system the defined performance can be achieved. Additionally, operational advantages that lead to a reduction of the handling effort can be achieved because transport and transshipment are functionally linked. The required investment costs can be estimated to be on a lower level compared to the ones of alternative 2, as in particular considerable cost benefits can be expected for the load carrier.

- **Evaluation of the application of underground transport systems (Alternative 6)**

  From an operational point of view the situation that occurs when underground transport systems are implemented, can be compared to the one of the above-mentioned alternative 2, at least concerning the basic structures of the operational processes and the vehicles concept. Moreover, one advantage is the underground alignment as this one has the least environmental impact and, therefore, a greater (political) enforceability. Independently from the actual extent of resistance, there are significant technical and operational problems. For instance, once an alignment is determined an adaptation to changing structures will entail enormous costs. Apart from that maintenance and overhauling work as well as measures taken when it comes to an operational disturbance (for example in case of accidents) are considerably more time- and cost-consuming whereby it might come to related system downtimes. In addition, it is necessary to keep the comparatively high investment costs for such an infrastructure in mind.

  The operational processes are substantially the same as those of alternative 2. Disadvantages appear concerning the associated investment costs and the lacking flexibility in the case of spatially changing demand structures.

To compare the six alternatives four *main criteria* are chosen, the *operational performance*, the *technical realizability*, the resulting *investment* and *operating costs*, as well as the *political enforceability* (referring to the German situation). Starting point are the technical and capacity requirements described in Section 4, because these determine the framework of the underlying discussion. The information is drawn either from technical studies or statements of manufacturers of (heavy-lift) transportation systems. The results are presented in Tab. 1.

Table 1. Comparison of the alternatives

	(1) Shuttle trains	**(2)** Guided vehicles on elevated tracks	**(3)** Cable car system	**(4)** Aerial ropeway system	**(5)** Overhead conveying system	**(6)** Underground transport systems
Operational performance	-	+	--	--	+	+
Technical realizability	+	+	(n.p.)	(n.p.)	+	-
Costs	-	0	-	-	+	-
Political enforceability	-	+	-	0	0	+

Ratings: ++ Excellent / + Good / 0 Indifferent / - Not so good / -- Poor; n.p. → not possible

Apart from the technical and operational aspects presented for each of the different alternatives it is fundamental to find out the time of availability. Infrastructural measures to this extent require a larger time frame from planning up to legally binding approval as well as a correspondent time period for realizing the measures. Evaluating the situation from a realistic point of view taking into account past experiences with large-scale infrastructure projects (for example in Germany), it is indeed doubtful whether a realization within a period of ten years (or less) is possible.

6 CONCLUSION AND PROSPECTS

Even if the presented alternatives are basically feasible, it is not yet the question which of these alternatives should be implemented. More important is the development of an integrated concept with a catalogue of measures in order to be able to react effectively on the increasing amount of containers in the next years. In this context it has to be questioned whether future demands will rise to dimensions of up to ten million TEU. Moreover, it is necessary to bear in mind the percentage of transit containers (in the feeder traffic) as well as the one of local clients, for which it makes no sense to handle these within an ST. Additionally, there are considerations about shifting container transport to the inland waterway network in order to use inland ports with sufficient capacities as transshipment facilities to offer an alternative to rail or road freight transport. Apart from the actual amount of containers to be handled in an ST it is inevitable to relieve the SCT of Hamburg‘s port, that means, corresponding methods are necessary in order to avoid any capacity bottlenecks in the medium and long-term future.

At this stage the aspect of time plays an important role. Implementing infrastructural measures usually takes some time because of the rights of directly affected people to appeal against these measures (partly in connection with long court proceedings) but also due to objections on the political level (which partly might have opportunistic reasons). Apart from that there is the necessity to evaluate the possible risks. If the focus is laid on only one alternative a strong (and under certain circumstances a high-risk) dependence will arise, so that in the case of operational disturbances rapidly bottlenecks occur that might not be tackled. The resulting risks might be enormous and even lead to a total breakdown of operations within the SCT (as the worst-case). Apart from the risk management, shuttle connections should not be seen as an isolated measure but as part of a list of measures, which also includes changes in the existing infrastructure and all activities within the SCT.

Considering the different aspects, a reasonable answer seems to be a *hybrid solution* with temporal, technical and operational facets. The latter, realigning itself to the developing demand structures, is bound to include a forced extension of rail freight transport connections, whereas the use of the existing rail / rail-transshipment facilities is mandatory for a more efficient design of container flows from and to the hinterland. If it comes to a Europe-wide implementation of a *MegaHub network* on the long run (see, e.g., Limbourg and Jourquin 2009), further chances for advancement would appear, especially concerning the shipping times. Simultaneously to these measures it is essential in the medium term (or rather on the long run) to install a sufficiently dimensioned high-capacity shuttle system which is able to cope with the expected transport demand. Moreover, in context with related risk aspects system design needs to include precautionary measures to be highly robust against partial system failures which possibly may appear. Based on the high-performance requirements as well as the technical and economical framework, alternatives 2 and 5 seem at first glance to be the most reasonable solutions. A detailed examination about which of these solutions should be eventually implemented is essential. Alternative 6, that is largely equal to 2 and 5 regarding the operational processes, leads however, to high investment costs and a long-term static network structure which can only be adapted to changing needs with enormous costs.

With regards to the expected development of the container throughput for the port of Hamburg it is not the question whether it is reasonable to implement an ST, but simply how to proceed with the implementation. As mentioned above the period of time for the necessary measures is estimated to last for several years, the fundamental decisions have to be made promptly. Probably in the medium, but at least on the long run the competitiveness with other seaports of the north range like Rotterdam or Antwerp will be influenced negatively, if capacity bottlenecks occur which result in incalculable disruptions within the operational processes.

REFERENCES

Alicke, K. (2002), Modelling and optimization of the intermodal terminal Mega Hub, *OR Spectrum*, **24**, 1-17.

Arnold, D. and Rall, B. (1996), Neues Umschlagsystem für Güterverkehrszentren im Vergleich mit aktuellen Konzepten, In Verein Deutscher Ingenieure (VDI) (ed.), *Innovative Umschlagsysteme an der Schiene*, VDI, Düsseldorf, 197-208.

Bontekoning, Y.M. (2006), *Hub exchange operations in intermodal hub-and-spoke networks*, Proefschrift Technische Universiteit Delft

Braet, J. (2011), The environmental impact of container pipeline transport compered to road transport - Case study in the Antwerp Harbor region and some extrapolations, *International Journal of Life Cycle Assessment* **16**, 886-896.

Daduna, J.R. (2011), Importance of hinterland transport networks for operational efficiency in seaport container terminals, In J.W. Böse (ed), *Handbook of Terminal Planning,* Springer, New York, 381-397.

Dellmann, T. and Berger, A. (2006), Autonomie und Automation der Eisenbahn im Kontext der Verkehrstechnik, *ETR Eisenbahntechnische Rundschau* **55**, 233-240.

Dimitrijevic, B. and Spasovic, L.N. (2006), Providing linkages between port terminals and inland freight distribution facilities. Paper presented at the METRANs National Urban Freight Conference, Long Beach, CA, Feb 01-03, 2006
http://www.metrans.org/nuf/documents/dimitrijevic.pdf, last access Feb 8, 2012.

Franke, K.-P. (2008), A technical approach to the agile port system, In R. Konings, H. Priemus, and P. Nijkamp (eds), *The future of intermodal freight transport*, Edgar Elgar Publishing, Cheltenham (UK), 135-151.

James, K. (2008), Assessing near-dock rail loading and offloading procedures at the ports of LA/LB for application to a container conveyor to ICTFs. Final Report METRANS Project

07-323306, http://www. metrans.org/research/final/AR-06-02-Final-Report.pdf, last access Feb 8, 2012.
Limbourg, S. and Jourquin, B. (2009), Optimal rail-road container terminal locations on the European network. *Transportation Research E* **45**, 551-563.
Marnette, W. and Böwer, T. (2011), Container & Underground Hamburg. Ein Infrastruktur- und Wachstumsprojekt für Hamburg und die Metropolregion. Slides Prefeasibility Study, August 2011.
http://image1.ffn.de/fileadmin/galerien/aktuell/container_gutachten.pdf, last access May 31, 2012: 23.01.2012.
Pallasch, A.K., Piworus, S., and Rickert, L. (2010), Eine Seilbahn für Container, *Hebezeuge und Fördermittel* **50**, 254-255.
Pielage, B.-J. and Rijsenbrij, J. (2005), Developments in underground freiht transport, In R. Konings, R., H. Priemus, and P. Nijkamp (eds), *The future of intermodal freight transport*, Edgar Elgar Publishing, Cheltenham (UK), 65-83.
PLANCO Consulting GmbH (2007), *Prognose der deutschlandweiten Verkehrsverflechtung - Seeverkehrsprognose* (Los 3). Gutachten im Auftrag des Bundesministeriums für Verkehr, Bau und Stadtentwicklung
Rodrigue, J.-P. (2008), The Thruport concept and transmodal rail freight distribution in North America, *Journal of Transport Geography* **16**, 233-246.
Roop, S.S., Ragab, A.H., Olson, L.B., Protopapa, A.A., Yager, M.A., Morgan, C.A., Warner, J.E., Mander, J., Parkar, A., and Roy, S.L. (2011), The freight shuttle system - Advancing commercial readiness. Technical Report 9-1528-1, Texas Transportation Institute, The Texas A&M University, College Station, http://www.tti.tamu.eda/documents/9-1528-1.pdf, last access Feb 7, 2012.
Rosa, A. and Roscelli, R. (2009), Innovative ideas and design of an integrated dry port and sea port system, In United Nations Economic and Social Commission for Asia and the Pacific (ESCAP) (ed), *Transport and Communications Bulletin for Asia and the Pacific* **78** (Development of Dry Ports), 57-72.
Roso, V. (2008): Factors influencing implementation of a dry port, *International Journal of Physical Distribution and Logistics Management* **38**, 782-798.
Roso, V., Woxenius, J. and Lumsden, K. (2009), The dry port concept - Connecting container seaports with the hinterland, *Journal of Transport Geography* **17**, 338-345.
Schönemann, R. and Plattner, T. (2012): Innovative freight transport systems for ports and terminals, In P. Golinska, and M. Hajdul (eds), *Sustainable transport*, Springer, Berlin, 295-315.
Schwartz, F., Stahlbock, R., Voß, S. (2009) Evaluation of postponement activities in dry ports. In: *Proceedings of the IAME 2009 Conference*, IAME, Copenhagen, Paper 2-77, 19 pages.
Slack, B. (1999), Satellite terminals - A local solution to hub congestion?, *Journal of Transport Geography* **7**, 241-246.
United Nations Conference on Trade and Development (UNCTAD) Secretariat (ed) (2009), *Review of Maritime Transport*, United Nations Publication, New York.
United Nations Conference on Trade and Development (UNCTAD) Secretariat (ed) (2010), *Review of Maritime Transport*, United Nations Publication, New York.
United Nations Conference on Trade and Development (UNCTAD) Secretariat (ed) (2011), *Review of Maritime Transport*, United Nations Publication, New York.
Visser, J., Wiegmans, B.W., and Konings, R. (2008), Review of underground logistic systems in the Netherlands - An ex-post evaluation of barriers, enablers and spin-off. Paper presented at the 5th International Symposium on Underground Freight Transportation by Capsule Pipelines and Other Tube/Tunnel Systems (ISUFT 2008) Arlington, March 20-22.

META-HEURISTICS FOR A MULTI-PRODUCT DYNAMIC LOT-SIZING AND SHIPPING PROBLEM WITH MULTIPLE FREIGHT CONTAINER TYPES

Byung Soo Kim[1] , Woon-Seek Lee[2] and Cheol Min Joo[3]

[1]*Graduate School of Management of Technology, Pukyong National University 45, Yongso-ro, Nam-Gu, Busan 608-737, Korea, iekbs@pknu.ac.kr*

[2]*Department of Systems Management & Engineering, Pukyong National University 45, Yongso-ro, Nam-Gu, Busan 608-737, Korea, iewslee@pknu.ac.kr*

[3]*Department of Industrial & Management Engineering, Dongseo University San 69-1, Ju-re, Sa-sang, Busan, Korea, cmjoo@dongseo.ac.kr*

ABSTRACT

This paper analyzes a dynamic lot-sizing problem, in which the order size of multiple products and multiple container types are simultaneously considered. In the problem, each ordered products placed in a period is immediately shipped by multiple freight containers in the period. Moreover, each container has type-dependent carrying capacity restriction. The unit freight cost for each container type depends on the size of its carrying capacity and the total freight cost is proportional to the number of each container type employed. Also, it is assumed that the load size of each product is equal and backlogging is not allowed. The objective of this study is to simultaneously determine the lot-sizes and the shipment schedules that minimize the total costs, which consist of setup cost, inventory holding cost, and freight cost. Because the problem is NP-hard, we propose three meta-heuristic algorithms with a local search heuristic, simulated annealing (SA) and genetic algorithm (GA), and a new population-based evolutionary meta-heuristic called self-evolution algorithm (SEA). The performances of the meta-heuristic algorithms are evaluated from a set of simulation experiments.

Keywords: Dynamic lot-sizing, Multi-products, Multi-freight container types, Meta-heuristics

1 INTRODUCTION

The manufacturing companies in a supply chain have been strategically allies with a specialized third party logistics (TPL) provider to reduce logistics costs related with transportation and warehousing, and they focus on their core business. In a view of the TPL warehousing provider, lot-sizing problem and inbound shipment scheduling in a dynamic time period is important issue to reduce transportation and warehousing costs. The Dynamic Lot-Sizing Model (DLSM) has stemmed from the work of Wagner and Whitin (1958). The majority of DLSMs have not considered any production-inventory problem incorporating transportation activities. These days, the issue of shipment scheduling for products (or delivering orders) by proper shipping modes at right time becomes significantly important in production (or distribution) management, or in import and export activities. Each manufacturing company produces multiple products and uses various types of freight containers as a shipping unit to deliver its manufactured products to customers. It may lead to the managerial decision problems: production rates for each product, container types used, loading policy in containers, and the number of containers used, etc. The decisions provide us with a motivation to consider the optimal lot-sizing and shipping problem incorporating production-inventory and shipment processes together.

Several articles have studied the extended works of the classical DLSM. Lippman (1969) studied two deterministic multi-period production planning models; monotone cost model and concave model. Hwang and Sohn (1985) dealt with a DLSM in which the transportation mode and the order size for a deteriorating product were simultaneously considered. However, they considered no capacity restriction on the transportation mode. Lee (1989) considered a DLSM allowing multiple setup costs including a fixed charge cost and a freight cost, where a fixed single container type with limited carrying capacity is considered and the freight cost is proportional to the number of containers used. Anily and Tzur (2005) considered a dynamic model of shipping multiple items by capacitated vehicles. They presented an algorithm based on a dynamic programming approach. Norden and Velde (2005) dealt with a multiple product problem of determining transportation lot-sizes in which the transportation cost function has piece-wise linear as to a transportation capacity reservation contract. Lee *et al.* (2005) proposed a heuristic algorithm for a dynamic lot-sizing problem and shipping scheduling, in which the order size of multiple products and a single container type are simultaneously considered. Kim and Lee (2012) proposed a shortest path reformulation model for lot-sizing and shipment scheduling problem in which multiple products with a single freight container type are considered and they is proposed to obtain a good lower bound in multi-product dynamic inbound ordering and shipment scheduling problem.

This paper analyzes a dynamic lot-sizing problem, in which the order size of multiple products and multiple freight container types are simultaneously considered. In the problem, each order (product) placed in a period is immediately shipped by multiple freight containers in the period. Moreover, each container has type-dependent carrying capacity restriction. The unit freight cost for each container type depends on the size of its carrying capacity and the total freight cost is proportional to the number of each container type employed. Also, it is assumed that the load size of each product is equal and backlogging is not allowed. The problem extends the work of Kim and Lee (2012) by considering multiple freight container types allowed. The objective of this study is to propose efficient heuristic algorithms to simultaneously determine the lot-sizes and the shipment schedules that minimize the total costs, which consist of setup cost, inventory holding cost, and freight cost.

In the next section, the mathematical model of the problem is described. In section 3, the properties of the optimal solution are characterized. Three meta-heuristic algorithms with a local search algorithm are proposed, based on the optimal solution properties in section 4. The computational results from a set of simulation experiment are presented in section 5.

2 MODEL FORMULATION

Some other notations are introduced as follows:

T = length of the time horizon,
t = time index ($t = 1,2,\cdots,T$),
M = number of products,
i = product index ($i = 1,2,\cdots,M$),
N = number of available freight container types,
j = index of container type ($j = 1,2,\cdots,N$),
d_{ti} = amount demanded for product *i* in period *t*,
$M_{ti} = \sum_{k=t}^{T} d_{ki}$,
W_j = carrying capacity of container type *j*,

x_{tij} = amount of product i that produced (ordered) and shipped by container type j in period t,

y_{tj} = number of container type j used in period t (nonnegative integer),

I_{ti} = amount of inventory of product i at the end of period t,

S_i = setup cost of product i in period t,

h_i = unit inventory holding cost of product i from period t to period t+1,

F_j = unit freight cost of container type j in period t, and

z_{ti} = 1, if a setup is incurred for i in t, and 0, otherwise.

The proposed problem has the objective determining a set of decision variables $(\sum_{j=1}^{N} x_{tij}, y_{tj}, z_{ti})$ for $t = 1,2,\cdots,T$, $i = 1,2,\cdots,M$, and $j = 1,2,\cdots,N$ such that all the demands over the given horizon are satisfied at the minimum total cost. Therefore, the T-period problem can be formulated in an integer programming as follows:

$$(P) \qquad \underset{x_{tij}, y_{tj}}{Minimize} \sum_{t=1}^{T} \{ \sum_{i=1}^{M} S_i \cdot z_{ti} + \sum_{i=1}^{M} h_i \cdot I_{ti} + \sum_{j=1}^{N} F_j \cdot y_{tj} \} \tag{1}$$

$$s.t. \qquad I_{ti} = I_{t-1,i} + \sum_{j=1}^{N} x_{tij} - d_{ti}, \qquad \forall t, i, \tag{2}$$

$$\sum_{i=1}^{M} x_{tij} \le W_j \cdot y_{tj}, \qquad \forall t, j, \tag{3}$$

$$\sum_{j=1}^{N} x_{tij} \le M_{ti} \cdot z_{ti}, \qquad \forall t, i, \tag{4}$$

$$I_{0i} = I_{Ti} = 0, \qquad \forall i, \tag{5}$$

$$x_{tij} \ge 0, \; I_{ti} \ge 0, \qquad \forall t, i, j, \tag{6}$$

$$y_{tj}: \text{nonnegative integer}, \qquad \forall t, j. \tag{7}$$

The objective function (1) minimizes the total cost of setup cost, holding cost, and freight cost for all products shipped by multiple container types over the entire time horizon. Constraint (2) is used to derive the balance of inventories between consecutive periods. Constraint (3) guarantees that the total production and shipping amount is restricted by the total carrying capacity associated with each vehicle type in a period. Constraint (4) ensures that the correlation between x_{tij} and z_{ti}. A closed bounded convex set is defined based on all the constraints and the objective function is concave, so that it attains its minimum at an extreme point of the convex set. In the next section, the extreme points shall be characterized further in association with the optimal solution.

3 OPTIMAL SOLUTION PROPERTIES

The mathematical model P can be represented by a network model as Figure 1. In the network, two flow types are defined as follows:

(1) The aggregate flow is defined as the flow between node 0 and nodes $(1,2,\cdots,T)$.

(2) The individual flow is defined as the flow between nodes $(1,2,\cdots,T)$ and nodes ((1,1), $(1,2),\cdots,(T,M))$.

Here, the arcs in the aggregate flow are restricted by the capacities associated with the number of containers used whereas the arcs in the individual flow are not.

The optimal solution of the model *P* occurs at extreme points. In a network theory, such an extreme point can be interpreted as an extreme flow (refer to Florian *et al.*, 1980; Zangwill, 1969). In a network without arc capacities, a feasible flow is an extreme flow if it does not have a positive loop. Also, in a network with arc capacities, a feasible flow is an extreme flow if and only if each loop has at least one saturated arc.

In Figure 1, loops can be formed by three cases as follows:

(1) Between the aggregate flows, for example, the loop can be formed by the sequences of nodes (0), (1), and (0).
(2) Between the aggregate flow and the individual flow, for example, the loop can be formed by the sequences of nodes (0), (1), (1,1), (2,1), (2), and (0).
(3) On the individual flows, for example, the loop can be formed by the sequences of nodes (1,1), (1), (1,*M*), (2,*M*), (2), (2,1), and (1,1).

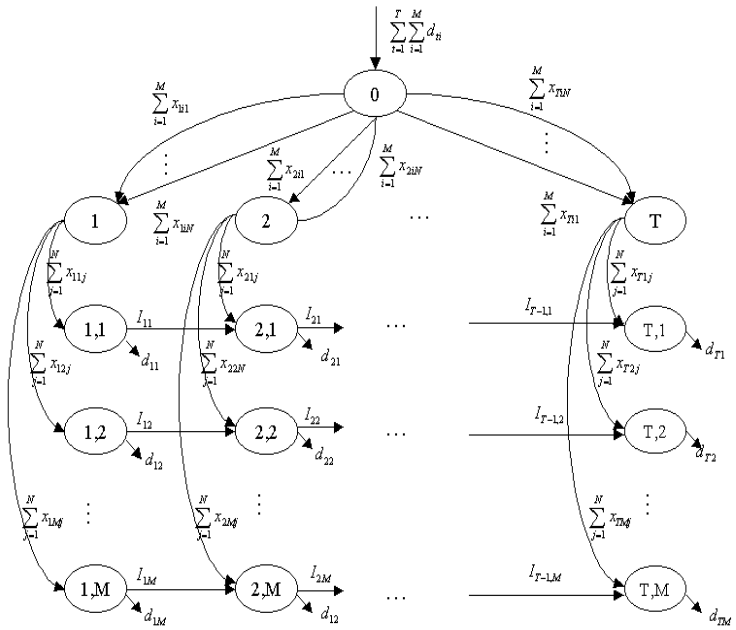


Figure 1. Network representation of the model *P*

Production and shipment schedule satisfying the optimal solution property of Wagner and Whitin (1958), $\sum_{j=1}^{N} x_{tij} \cdot I_{t-1,i} = 0$, is the extreme flow no longer. Such policies may form positive loops between the aggregate flows. The properties of theorem 1, 2, and 3 must be satisfied so as to have an extreme flow. To examine the properties, the production point, the partial aggregate point, and the inventory point are defined, respectively as follows:

(1) Period t is a production point for product i if $\sum_{j=1}^{N} x_{tij} > 0$.

(2) Container type j is a partial aggregate point in period t if $nW_j < \sum_{i=1}^{M} x_{tij} < (n+1)W_j$, where n is a non-negative integer.

(3) Period t is an inventory point for product i if $I_{ti} = 0$.

(4) Period t is a partial aggregate point for product i if period t is a production point for product i and there are some partial containers in period t.

Theorem 1. *In the model P, the optimal solution has at most one partial container in any period t.*

Proof. Suppose that there exists the optimal solution that has two partial containers 1 and N in period 1. In the network of Figure 1, this case has the positive loop formed by the sequences of nodes (0), (1), and (0). Two arcs in the loop are restricted by the carrying capacities of the container types 1 and N, respectively. This feasible flow is not an extreme flow because no arcs are saturated. Therefore, the proof is completed. □

Theorem 2. *In the model P, the optimal solution has at most one partial aggregate point for product i between two consecutive inventory points for product i.*

Proof. Suppose that there exists the optimal solution which has two partial aggregate point between two consecutive inventory points for product *i*. In the network of Figure 1, this case can have the loop formed by the sequences of nodes (0), (1), (1,1), (2,1), (2), and (0). The condition that this feasible flow will be the extreme flow is that at least one of arcs (0,1) and (0,2) must be saturated. This feasible flow is not an extreme flow. Therefore, the proof is completed. □

Theorem 3. *In the model P, the optimal flow must do not form the positive loop in the individual flow.*

Proof. Suppose that there exists a feasible flow that satisfy the properties of Theorem 1 and 2 , which has a loop formed by the sequences of nodes (1,1), (1), (1,2), (2,2), (2), (2,3), (3,3), (3), (3,1), (2,1), and (1,1) in Figure 1. Because arcs in the individual flow are not capacitated, there exists a positive loop formed by unsaturated arcs. This feasible flow is not an extreme flow. Therefore, the proof is completed. □

Unfortunately, the problem *P* is NP-hard. So, it is not easy to make optimization viable for large problems. Therefore, three meta-heuristic algorithms are presented based on the properties of Theorem 1, 2, and 3 in the next section.

4 META-HEURISTICS

In this section, we propose three meta-heuristics, simulated annealing (SA), genetic algorithm (GA), and self-evolution algorithm (SEA) to determine lot-sizing of each product and container types to be shipped in each period. The heuristics have two-phase procedure. In the first phase, the decision of lot-sizing is provided by the meta-heuristics and then the decision of container types for shipment is determined by a local-search based on aggregating the decision of lot-sizing of each product.

4.1 Solution Representation

For the solution representation of three meta-heuristic algorithms, a solution is defined as two dimensional array of (*T*, *M*) of 0-1 genes as follows:

$$\boldsymbol{E} = \begin{bmatrix} e_{11} & \dots & e_{T1} \\ \vdots & \ddots & \vdots \\ e_{1M} & \dots & e_{TM} \end{bmatrix} \tag{8}$$

In this representation, $\boldsymbol{E}$ has the perfectly same meaning of variable $\mathbf{z}$ in Section 2, where e_{tm} can take a value 0, or 1, which is the key to find the value of $\sum_{j=1}^{N} x_{tij}$. In this situation, the value of e_{ti} decides the value of $\sum_{j=1}^{N} x_{tij}$ in each period. If e_{ti} =0, there is no production of product *i* at period *t*, i.e., $\sum_{j=1}^{N} x_{tij}$ =0. If e_{ti} =1, the value of productions which covers d_{Vi} where *V* is the set of current and subsequent periods of the period *t* with corresponding e_{Vi} =0. In this representation, one can easily find that the first nonzero value of e_{ti}, t = 1, 2, …, *T*, must be '1' no later than the first nonzero value of d_{ti} to meet Constraint (6) indicating a backlogging is not allowed. Thus, a repairing process requires after genetic operations of GA and SEA and perturbations of SA.

4.2 Local Search Heuristics

Once aggregate production X_t is calculated by $\sum_{j=1}^{N} x_{tij}$, z_{ti} and y_{tj} are determined by a local search heuristic using *the marginal cost heuristic per unit-product*. The detailed algorithm for *the marginal cost heuristic per unit-product* is described as follows.

Step 1: Calculate the unit-freight cost, $M_j = F_j / W_j$, for $\forall j$.

Step 2: Select the container type *p* with the lowest M_p and assign $y_{tp} = \lfloor X_t / W_p \rfloor$ and select the container type q with the lowest F_q and $y_{tq} = 1$.

Step 3: Set t=1.

Step 4: Set product list $P = \{1, 2, \dots, M\}$.

Step 5: If $P = \phi$, $t = t+1$ and go to *Step 4*. Otherwise, select product *k* with the highest h_k. Assign $z_{tk} = 1$.

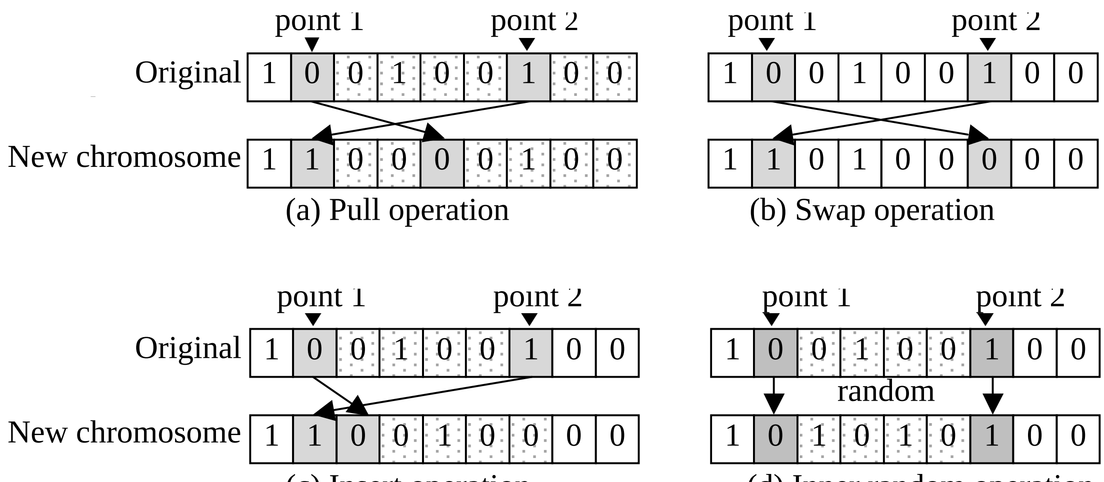
Step 6: Update $X_t = X_t - d_{tk}$ and $d_{tk} = 0$ and eliminate *k* from P.

Step 7: If $X_t < W_q$ with q is the lowest value of $|W_j|$, $j = \{1, \dots, N\}$, go to *Step 5*. Otherwise, select the container type r with the lowest F_r and product r with the highest h_r and set $z_{tr} = 1$.

4.3 Genetic Algorithm

The genetic algorithm (GA), which has been widely used in various optimization problems for last three decades, is a stochastic search algorithm based on the mechanism of natural selection and natural genetics. Being different from conventional search techniques, it starts with an initial set of (random) solutions called a population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. The chromosomes are evaluated, using some measures of fitness. Generally speaking, the genetic algorithm is applied to spaces that are too large to be exhaustively searched (Goldberg, 1989). In this paper, the chromosomes that have higher fitness value (lower objective function value) than the average fitness of current population make a potential parent pool and new chromosomes in the next generation are randomly selected from the parent pool or selected from the roulette wheel method.

Two kinds of crossover operators which are PMX crossover and Uniform crossover are randomly selected and eight kinds of mutation operators the eight mutation operators which are pull operator, insert operator, swap operator, inner random operator and outer random operator are randomly selected to make a new chromosome. For the mutation operators, one, two or multiple points in the selected original chromosome are randomly selected. The pull operator is illustrated in Figure 2 (a). The genes on right side of point 2 (including point 2) are pulled to the position of point 1, and the genes between point 1 and 2 (including point 1) are placed after. The two genes at the points are interchanged for swap operator as shown in Figure 2 (b). Insert operator simply insert the gene at point 2 into the position of point 1 as shown in Figure 2 (c). Inner random operator and outer random operator are illustrated in Figure 2 (d) and (e). The inner or outer genes of point 1 and 2 are randomly replaced for the operators. Uniform random operator randomly select multiple points and change as shown in Figure 2 (f). Figure 2 (g) and (h) describe single change and other change in simple way. Using the crossover operators, the mutation operators, and the selection operators, the selected parents reproduce new chromosomes (i.e., children) to generate a population for the next generation. The various parameters of the GA heuristic are summarized in Table 1. These parameters are selected based on extensive preliminary experimentations to the best combination with highest performance. GA evaluates a total of 1000 × (T+M+N) of fitness function.



(a) Pull operation (b) Swap operation

(c) Insert operation (d) Inner random operation

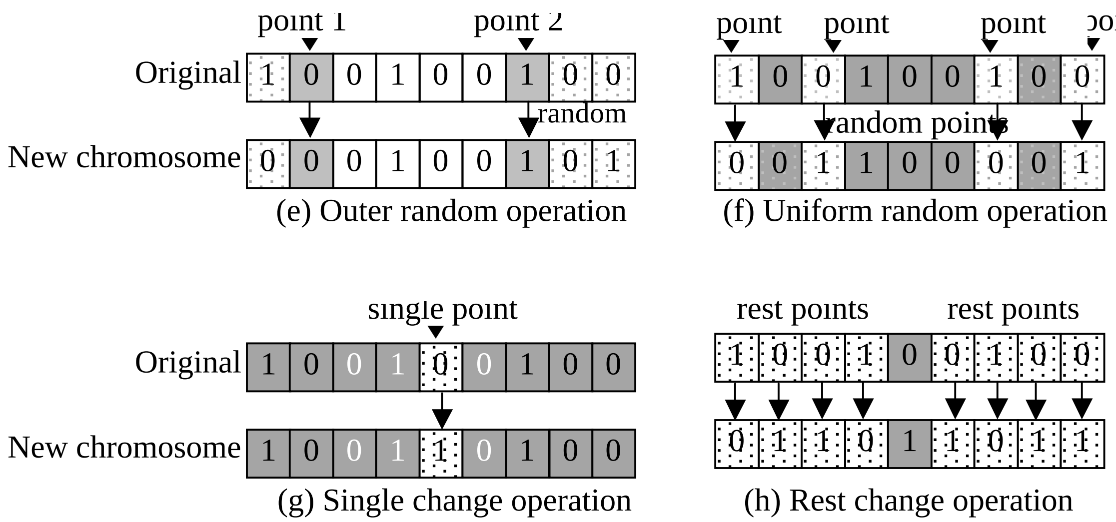


Figure 2. Mutations of GA

Table 1. Parameters of the GA heuristics

Parameter	Value/type
Population size	$T+M+N$
Number of generations	1000
Crossover rate	0.8
Mutation rate	0.2

4.4 Simulated Annealing

The simulated annealing (SA) was first proposed by Kirkpatrick *et al.* (1983) to solve combinatorial optimization problems. SA explores the solution space by successive moves from one potential solution to another that is a variation of its predecessor. At the start of the process, an initial solution is generated and evaluated based on the total cost (fitness) in Section 2.s. SA differs in that the procedure uses random selection and will sometimes accept non-improving moves (interchanges) hoping to expand the search space and ultimately reach a better overall solution. The non-improving moves are probabilistically performed using the Boltzman probability mass function as follows (Wolsey 1998):

$$p(T) = \exp(-\Delta f / T), \tag{9}$$

where T is the current temperature, $\Delta f = f(P_n) - f(P_c)$, and P_n and P_c are the candidate fitness and the current fitness values, respectively.

In the implementation of this paper, SA heuristic starts with a randomly generated chromosome and is controlled using two loops; an outer loop and an inner loop. The outer loop controls the temperature reduction according to the cooling schedule. In the inner loop, the temperature is held constant and a predetermined number of explorations are made. In the outer loop, explorations are made at each temperature. For fair comparison with other heuristics, the number of inner loops is 50 and the number of outer loops is 20, and the number of perturbations consists of (T+M+N) using random selection of the eight mutations in Figure 2 to increase the chances of obtaining a better solution. This leads a total of 1000 × (T+M+N) evaluations. This high number of perturbations was motivated by the research of Barbarosoðlu and Özdamar (2000) that indicated that the increase in the number of search moves carried out by SA significantly improves its performance. The various parameters of the SA heuristic are summarized in Table 2. These parameters were selected based on extensive preliminary experimentations to determine the best combination that leads to the highest frequency of hitting the optimal solution.

Table 2. Parameters of the SA heuristics

Parameter	Value/type
Initial Temperature	1000
Cooling Schedule	Logarithmic
Size of outer loop	20
Size of inner loop	50
Number of perturbations	T+M+N

4.5 Self-Evolutionary Algorithm

The self-evolutionary algorithm (SEA) is a meta-heuristic algorithm which has a population (a set of solutions) based mechanism using the evolution of a solution by itself (self-evolution). Similar to GA, the set of chromosomes forms a population. Initial population is generated randomly, and the chromosomes in the population are evaluated by the measure of the fitness introduced in Section 2. A chromosome from the population is randomly selected and executes a self-reproduction using a randomly selected evolution operator to make a new chromosome. Then the new chromosome is evaluated and it replaces the original chromosome, if the fitness value of the new chromosome is better than that of the original chromosome. The algorithm continues until the number of self-reproductions becomes a predetermined stopping value.

We propose the operators used in the eight mutations in GA. SEA is running without providing any parameters for the algorithm, because all the selection processes in SEA, such as selection of chromosome from the population for self-evolution, selection of evolution operator, and selection of points for the operator are randomly executed. For fair comparison with other heuristics, the number of generations 1000, and the number of evolution operators consists of (T+M+N) using random selection of the eight mutations in Figure 2 to increase the chances of obtaining a better solution. This leads a total of 1000 × (T+M+N) evaluations with the same number of evaluations of GA and SA. The various parameters of the SEA heuristic are summarized in Table 3. These parameters were selected based on extensive preliminary experimentations to determine the best combination that leads to the highest frequency of hitting the optimal solution.

Table 3. Parameters of the SEA heuristics

Parameter	Value/type
Operator size	T+M+N
Number of generations	1000

5 COMPUTATIONAL RESULTS

To analyze the performance of the proposed heuristic algorithm, the following experimental conditions were designed.

(1) Set M = 3, 6, 10 and T = 12, 15, 21.

(2) Demands were generated from a normal distribution $N(\mu_i, \sigma_i^2)$.

(3) Mean μ_i was generated from an uniform distribution U(25, 100).

(4) Standard deviation σ_i was equally likely selected from μ_i or $\mu_i / 5$.

(5) Setup cost was selected as follows:

$S_i = TS_i^2 \cdot \mu_i / 2$ and TS_i =1, 3, 6, where TS_i denotes EOQ time supply.

(6) $h_i = 1$ was assumed without loss of generality.

(7) Set W_j = 100, 200, 300 and respective unit freight cost was selected as follows:

$F = i \cdot W_j$, i = 1, 2, 3.

To evaluate the performance of the heuristic, the C# computer code for the proposed heuristic was run on an Intel Core™2 Duo CPU with 2.00GHz RAM. Also, CPLEX 12.2 package for finding the optimal solution was run on the same computer. But, the optimal solution was not obtained within 2 hours for many large-sized test problems having more than 8 periods because of the limitations of a computer performance.

Tables 4 and 5 shows relative percentage deviation (*RPD*) calculated with the expression (10) and computing time by GA, SA, and SEA of 10 replications for each test problem.

$$\frac{Z_H - Z_B}{Z_B} \times 100, \tag{10}$$

where Z_B = objective value of the best solution among the solutions of replications of GA, SA, and SEA and Z_H = objective value of the heuristic solution.

In Table 4, average *RPD*s of SEA, SA, and GA are 2.19, 12.34, and 21.92, respectively. We can see that SEA is the most effective algorithm with the lowest *RPD* and variation among the meta-heuristic algorithms among SA and GA. Meanwhile, average execution times per instance for GA, SA, and SEA in Table 5 were 5.52, 0.54, and 0.66 seconds, respectively. Thus, SA shows the lowest computing time, but the difference of the computing time between SA and SEA is not significant by showing less than 1 second. The computing time of three meta-heuristic algorithms increases, as *T*, *M*, and *N* increase.

Table 4. *RPD*s of GA, SA, and SEA

			T=12			*T*=15			*T*=21		
M	*N*	*W*	GA	SA	SEA	GA	SA	SEA	GA	SA	SEA
		100	16.06	6.11	0.79	18.51	11.04	1.25	12.30	6.88	0.64
	2	200	17.48	7.05	0.56	18.97	11.46	0.83	26.19	16.58	1.92
		300	40.99	21.75	2.81	26.40	10.38	0.34	23.73	11.29	0.69
		100	17.22	8.82	1.86	18.83	9.91	1.68	30.49	19.69	4.21
3	3	200	41.50	17.34	0.15	31.66	19.62	2.66	25.49	17.55	1.62
		300	5.97	2.78	0.08	21.16	6.04	0.31	28.49	15.98	1.43
		100	26.65	10.07	0.74	19.81	10.01	0.55	30.07	20.83	3.06
	4	200	16.78	6.31	0.65	13.08	3.47	0.08	22.70	6.29	0.26
		300	6.17	2.45	0.00	8.91	4.11	0.08	46.52	28.84	1.02
		100	16.34	10.63	1.18	10.87	6.96	1.14	16.34	10.03	0.55
	2	200	16.03	9.43	1.75	24.42	15.98	4.89	27.71	18.82	2.29
		300	23.48	14.07	3.43	21.57	14.29	4.22	27.22	17.68	3.57
		100	22.76	13.65	3.34	17.49	10.68	1.14	18.44	11.18	1.18
6	3	200	26.45	15.19	3.75	36.74	23.76	7.72	27.98	16.60	2.72
		300	27.65	16.76	4.61	30.02	17.54	5.09	27.91	17.69	4.64

		100	15.93	8.76	1.57	19.44	11.00	2.62	20.80	12.55	1.49
	4	200	28.95	16.06	3.45	34.35	17.06	2.24	29.78	15.81	3.24
		300	24.40	11.85	3.61	36.03	19.14	8.16	36.91	23.99	7.50
		100	9.46	5.72	0.21	8.48	5.63	0.66	12.12	7.82	0.21
	2	200	15.66	9.35	2.28	12.69	8.16	0.81	11.34	6.81	1.15
		300	19.03	10.42	1.80	19.84	11.47	2.05	20.89	13.84	1.47
		100	10.45	6.62	0.41	15.57	10.29	1.91	11.38	6.92	0.51
10	3	200	11.47	6.26	1.20	17.20	8.90	1.60	17.08	11.43	1.14
		300	25.90	14.41	3.89	28.11	17.17	6.33	23.11	13.63	1.54
		100	18.96	11.46	2.24	12.71	7.85	0.49	13.67	8.54	1.14
	4	200	33.26	17.51	7.21	27.74	13.36	2.08	17.72	11.15	2.10
		300	26.67	12.02	3.03	30.73	17.02	4.01	26.25	15.77	4.25
		Avg.	20.80	10.85	2.10	21.53	11.94	2.40	23.43	14.23	2.06

Table 5. Computing time of GA, SA, and SEA

	T=12	T=15	T=21	M=3	M=6	M=10	N=2	N=3	N=4
GA	2.47	4.15	9.94	1.65	4.35	10.56	5.20	5.59	5.96
SA	0.35	0.48	0.78	0.27	0.49	0.85	0.51	0.54	0.58
SEA	0.43	0.59	0.97	0.34	0.61	1.05	0.63	0.66	0.71

6 CONCLUSION

This paper analyzed a dynamic lot-sizing and shipping problem, in which the order size of multiple products and multiple container types were simultaneously considered. Because the problem is NP-hard, three meta-heuristic algorithms were proposed based on the optimal solution properties. To evaluate the performance of the heuristic, we propose three meta-heuristic algorithms with a local search heuristic, simulated annealing (SA) and genetic algorithm (GA), and self-evolution algorithm (SEA). SEA is a new meta-heuristic algorithm which has a population (a set of solutions) based self-evolution mechanism. *RPD* of SEA compared with the best solution is no more than 8.16 in worst case. In average sense, the test results indicate that SEA is an effective and efficient algorithm with low variation by showing 2.19 for a dynamic lot-sizing and shipping problem considering the order size of multiple products and multiple container types.

Further research will consider an extension of the problem where the loaded space size per unit for each product is heterogeneous and the production capacity is limited in each period.

REFERENCES

Anily, S. and Tzur, M., (2005), Shipping multiple items by capacitated vehicles: an optimal dynamic programming approach. *Transportation Science*, **39(2)**, 233-248.

Barbarosoðlu G and Özdamar L (2000) Analysis of solution space dependent performance of simulated annealing: the case of the multi-level capacitated lot sizing problem. *Computers and Operations Research*, **27(9),** 895–903

Florian, M., Rossin-Arthiat, M. and D. DeWerra (1971), A Property of Minimum Concave Cost Flows in Capacitated Networks, *INFOR*, **9**, 293-304.

Goldberg, D.E. (1989), *Genetic algorithms in search, optimization and machine learning*, New York: Addison-Wesley

Hwang, H. and Sohn, K. I. (1985), An Optimal Policy for the Dynamic Transportation-Inventory Model with Deteriorating Items, *IIE Transactions*, **17**, 233-241.

Kim, B. S. and Lee, W. S. (2012), A multi-product dynamic inbound ordering and shipment scheduling problem at a third-party warehouse, *International Journal of Industrial Engineering,* In press.

Kirkpatrick, S., Gellatt, C.D. Jr., and Vecchl, M.P. (1983), Optimization by simulated annealing, *Science*, **220**, 671-680.

Lee, C. Y., A (1989), Solution to the Multiple Set-Up Problem with Dynamic Demand, *IIE Transactions*, **21(3)**, 266-270.

Lee, W. S., Sox, C. R., Han, J. H., Cho, S. J. (2005), A Multi-Product Dynamic Lot-Sizing and Shipping Model, *International Journal of Production Economics*, **98**, 204-214.

Norden, L. van and Velde, S. van de, (2005), Multi-product lot-sizing with a transportation capacity reservation contract. *European Journal of Operational Research*, **165 (1)**, 127-138.

Lippman, S. A. (1969), Optimal Inventory Policy with Multiple Set-Up Costs, *Management Science*, **1(1)**, 18-138.

Wagner, H. M. and Whitin, T. M. (1958), Dynamic Version of the Economic Lot Size Model, *Management Science*, **5(1)**, 89-96.

Zangwill, W. I. (1969), A Backlogging Model and a Multi-Echelon Model of a Dynamic Lot Size Production System - a Network Approach, *Management Science*, **15(9)**, 506-527.

Wolsey, L.A. (1998), *Integer programming*, Wiley & Sons.

A SEQUENTIAL AND A SIMULTANEOUS SOLUTION APPROACH FOR A HINTERLAND CONTAINER TRANSPORTATION PROBLEM

Sebastian Sterzik[1] and Herbert Kopfer[1]

[1]*Chair of Logistics, Department of Economics and Business Studies, University of Bremen, Wilhelm-Herbst-Str. 5, 28359 Bremen, Germany, sterzik@uni-bremen.de, kopfer@uni-bremen.de*

ABSTRACT

In the hinterland of a seaport terminal, inbound as well as outbound containers have to be moved by a trucking company between the company's depot, the considered terminal and the company's customer locations. There are two types of customers. While the first customer type receives goods by inbound containers, the second customer type ships goods by outbound containers. Additionally, there are empty inbound containers and empty outbound containers available in the local area. For solving the problem one has to consider not only vehicle routing and scheduling but also empty container repositioning. This contribution presents a two-step model formulation as well as an integrated routing model formulation. Both formulations consider the routing of the means of transports as well as the routing of the transportation resources. Randomly generated test instances are employed and solved with the commercial tool CPLEX. The results show the effectiveness and efficiency of the sequential approach compared to the integrated routing solution approach.

Keywords: empty container repositioning, vehicle routing and scheduling, integrated routing, 2-step solution approach

1 INTRODUCTION

This contribution handles two different solution approaches for a comprehensive hinterland container transportation problem. The setting deals with full and empty containers which have to be moved by a trucking company in the hinterland region of a seaport terminal. The problem setting is based on Kopfer et al. (2011) who present an integrated exact solution approach performing the planning task of vehicle routing and scheduling and the task of allocating containers simultaneously. This integrated solution method is only able to solve small instances. Therefore, this contribution presents and analyzes an additional sequential solution approach which solves the container allocation problem separately on the first step and uses the obtained results as input for solving the vehicle routing and scheduling problem on the second step. Since sequential approaches are known to reduce the problem complexity at the expense of solution quality, it is expected that much less computation time is needed and, hence, bigger instances can be solved whereas the solution quality of small instances will decrease compared to the exact integrated approach. At first, this contribution determines the limitations of the 2-step approach in terms of the maximum size of the instances which are barely solvable. This sequential approach may lead to container allocations which are disadvantageous for the routing of the vehicles because they may require a gratuitous number of vehicles to move the employed containers. Thus, we introduce and test two different objectives for container allocation with regard to the impact on the vehicles' routing. Moreover, we compare the solution quality of the sequential and the simultaneous approach with each other and examine whether it is advisable to implement the 2-step approach heuristically.

We consider a hinterland transportation problem (see Figure 1) in which full and empty containers have to be moved between different locations by trucks. Thereby, a trucking company is responsible to serve its customer base with a homogeneous fleet of vehicles.

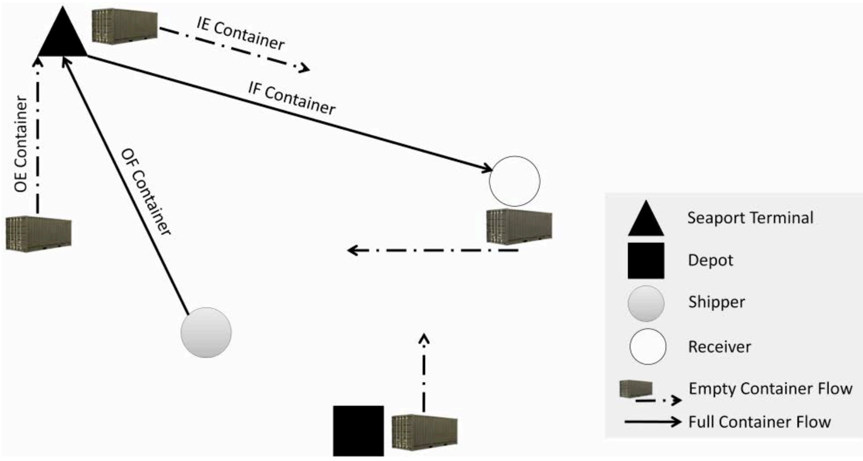


Figure 1. OD-CTTP (According to Kopfer et al. (2011))

We assume 40-feet-containers, i.e. a vehicle can only move one container at a time. The fleet is parked at the company’s depot in which, moreover, empty containers can be stacked. Each vehicle starts and ends its tour at the depot. Additionally, it is assumed that there is a seaport terminal to which the trucks carry full and empty containers from customers' places and vice versa. Time windows at the terminal as well as time windows at the customer nodes need to be taken into account. We distinguish two types of customers. On the one hand shippers offer freight which is to be transported to a foreign region via the seaport. The flow of a full container from the shipper to the terminal is defined as outbound full (OF) container. On the other hand receivers require the transport of their goods from an outside region via the terminal. The flow of a full container from the terminal to the receiver is called inbound full (IF) container. It is obvious that a shipper requires an empty container before he can fill the freight in it. Additionally, an empty container remains at the receiver's location after the IF container is unloaded. We define two time windows at each customer location: During the first time window the full/empty container has to be delivered to the customer location. After the container is unloaded/loaded, it can be picked up by a vehicle during the second time window. Moreover, the trucking company has to consider two additional types of transportation requests. Due to the imbalance between import- and export-dominated areas, they need to take care of outbound empty (OE) or inbound empty (IE) containers which either have to be moved to the terminal or derive from it. For these requests only the terminal as the destination or as the origin is given in advance. Hence, the locations that can provide empty containers for the OE requests and the destinations for the imported empty containers need to be determined during the solution process.

Since Zhang et al. (2009) defined a similar problem with several depots as the multi-depot container truck transportation problem (CTTP), the underlying problem is called one-depot container truck transportation problem (OD-CTTP). The objective is to minimize the carriers' total fulfilment costs consisting of fixed and variable costs. Hence, in a first step the number of used vehicles should be minimized while in the second step the optimization of the vehicles' total operating time symbolizing the transportation costs should be pursued (Toth and Vigo (2002)). In this paper the first objective is formulated as a constraint and the minimization of the operating time is chosen as the objective function of our models. In order to reach feasible solutions, the number of used vehicles within the employed models is raised iteratively until a feasible solution is found. Since not only the means of transports but also the transportation resources have to be considered, the resulting solution approaches lead to models which comprise vehicle routing and scheduling as well as the containers' flows. Thus, solving the problem determines to which locations empty containers should be delivered after they are released from an inbound full or empty load. Moreover, the locations at which empty container are picked up for outbound full or empty loads should be identified. Finally the implemented solution approaches need to determine in which order and by which vehicle the loads should be carried out.

2 MATHEMATICAL MODELS

The sequential method consists in the following two steps for solving the OD-CTTP. In the first step an optimal decision on the assignment of available empty containers to upcoming transportation tasks is aspired. This decision fixes an origin and a destination for each empty container which has to be transported. Since the containers' allocation is completely disconnected from the global objective (i.e. minimizing the costs of operating vehicles) and, moreover, since every container has to be moved by a vehicle, we want to avoid that the containers' allocation generates results which require a gratuitous amount of operating vehicles. Therefore, it is important to detect an adequate objective for the first step. Two different objective functions have been implemented and analyzed in terms of their impact on the solution space of the second step. The solution of the first step is then used as input for a modified version of the well known pickup and delivery problem with time windows (PDPTW; see e.g. Parragh et al. (2008)). Thus, the determined origins and destinations of the container transportation requests are used to find the best routes for the operating vehicles. The global objective is adopted for the modified PDPTW. The simultaneous method which is adopted from Kopfer et al. (2011) solves the two sub-problems of the sequential approach in one single step, i.e. solving the assignment problem of empty containers is done simultaneously with the vehicle routing and scheduling problem induced by the originally given problem data and the compulsory assignment decisions. It has to be guaranteed that the vehicles and the containers are interlinked with each other so that each container movement is enabled by a vehicle.

2.1 Node Sets, Variables and Parameters

The set of nodes V consists of customer node set V_C, terminal node set V_T and the depot location which consists of the start node *s(0)* and end node *e(0)*. The total number of customers is given by n whereas V_C is defined by the shippers $V_S = V_{S^i} \cup V_{S^o}$ and the receivers $V_R = V_{R^i} \cup V_{R^o}$. Thereby, nodes $i \in V_{S^i}$ and $(i+n) \in V_{S^o}$ as well as $j \in V_{R^i}$ and $(j+n) \in V_{R^o}$ mark the same location since node sets V_{S^i} / V_{R^i} and V_{S^o} / V_{R^o} define the first and second time window of the customers. Since for all IF and OF transportation requests, the pickup node and delivery node are explicitly given by the input data, every customer node has its corresponding terminal node. In case of an OF transportation request, this means that, after a shipper $i \in V_{S^o}$ has been served by a vehicle, the full container has to be moved to terminal node $(i+n) \in V_{T^{OF}}$. In case of an IF transportation request, a full container has to be moved from terminal node $i \in V_{T^{IF}}$ to its corresponding receiver location $(i-2n) \in V_{R^i}$. $V_{T^{IE}}$ and $V_{T^{OE}}$ illustrate the set of IE and OE containers which have to be transported. Therefore, V_T is defined through the inbound terminal node sets $V_{T^I} = V_{T^{IF}} \cup V_{T^{IE}}$ and the outbound terminal node sets $V_{T^O} = V_{T^{OF}} \cup V_{T^{OE}}$.

A vehicle $k \in K$ has to start its tour from node *s(0)* and end it at *e(0)*. As stated, we do not only regard the vehicles but also the containers as being transportation resources to be routed. The container set *C* is defined by the containers' possible starting locations. I.e. a container arises either from the terminal as an IF or IE container or it starts its path from the depot as an additional empty container. The service time s_i of node $i \in V_C \cup V_T$ can only be assured if the operating vehicle keeps the specified time window $[a_i / b_i]$. Thus, a vehicle has to arrive at location i before time b_i. However, arrival before a_i is allowed and leads to waiting time for the truck.

To sum it up, the following parameters, variables and node sets have to be defined:

$s(0)$:	Start node (Depot) of the vehicles
$e(0)$:	End node (Depot) of the vehicles
n:	Number of customers
m:	Number of vehicles
$[a_i / b_i]$:	Time window of node i
s_i:	Service time at node i
t_{ij}:	Travel time from node i to node j
M:	Sufficiently big constant
$V = V_C \cup V_T \cup \{s(0)\} \cup \{e(0)\}$	All node sets
$V_C = V_S \cup V_R$:	Customer node set
$V_S = V_{S^i} \cup V_{S^o}$:	Shipper node sets:
V_{S^i}:	- First time window
V_{S^o}:	- Second time window
$V_R = V_{R^i} \cup V_{R^o}$:	Receiver node sets:
V_{R^i}:	- First time window
V_{R^o}:	- Second time window
$V_T = V_{T^I} \cup V_{T^O}$:	Terminal node set
$V_{T^I} = V_{T^{IF}} \cup V_{T^{IE}}$:	Inbound terminal node sets corresponding to the IF and IE transportation requests
$V_{T^O} = V_{T^{OF}} \cup V_{T^{OE}}$:	Outbound terminal node sets corresponding to the OF and OE transportation requests
K:	Vehicle set
C:	Container set
x_{ijk}	1, if vehicle k drives from node i to j; 0 otherwise
y_{ijc}	1, if container c is moved from node *i* to *j*; 0 otherwise
T_{ik}	Represents the starting time of truck *k* from node *i*
L_{ic}	Represents the starting time of container *c* from node *i*

2.2 Sequential Approach

The optimization model for the first step (allocation of containers) consists in the equation (1) and the restrictions (2) to (17).

$$\min z1 = \sum_{i \in V_{T^I} \cup \{s(0)\}} \sum_{j \in V_{T^O} \cup \{e(0)\}} \sum_{c \in C} (L_{jc} - L_{ic}) \tag{1}$$

s.t.

$$\sum_{j \in V} \sum_{c \in C} y_{ijc} = 1 \qquad i \in V_C \cup V_{T^i} \tag{2}$$

$$\sum_{i \in V} \sum_{c \in C} y_{ijc} = 1 \qquad j \in V_{T^O} \tag{3}$$

$$\sum_{c\in C} y_{i(i-2n)c} = 1 \qquad i\in V_{T^{IF}} \qquad (4)$$

$$\sum_{j\in V_{S^i}\cup\{e(0)\}} \sum_{c\in C} y_{ijc} = 1 \qquad i\in V_{T^{IE}} \qquad (5)$$

$$\sum_{j\in V} \sum_{c\in C} y_{s(0)jc} = 1 \qquad (6)$$

$$\sum_{i\in V} \sum_{j\in V_{T^I}} \sum_{c\in C} y_{ijc} = 0 \qquad (7)$$

$$\sum_{i\in V_{T^O}\cup\{e(0)\}} \sum_{j\in V} \sum_{c\in C} y_{ijc} = 0 \qquad (8)$$

$$\sum_{i\in V} \sum_{j\in V_{T^O}\cup\{e(0)\}} y_{ijc} = 1 \qquad c\in C \qquad (9)$$

$$\sum_{i\in V_{\{s(0)\}}\cup V_{R^o}} \sum_{c\in C} y_{ijc} = 1 \qquad j\in V_{T^{OE}} \qquad (10)$$

$$\sum_{c\in C} y_{i(n+i)c} = 1 \qquad i\in V_{S^i\cup R^i} \qquad (11)$$

$$\sum_{j\in V} y_{jic} - \sum_{j\in V} y_{ijc} = 0 \qquad i\in V_C; c\in C \qquad (12)$$

$$L_{jc} \geq L_{ic} + t_{ij} + s_i - M(1-y_{ijc}) \qquad i,j\in V; c\in C \qquad (13)$$

$$\sum_{j\in V} y_{jic} * a_i \leq L_{ic} \leq \sum_{j\in V} y_{jic} * b_i \qquad i\in V_C \cup V_{T^O} \cup \{e(0)\}; c\in C \qquad (14)$$

$$\sum_{j\in V} y_{ijc} * a_i \leq L_{ic} \leq \sum_{j\in V} y_{ijc} * b_i \qquad i\in V_{T^I} \cup \{e(0)\}; c\in C \qquad (15)$$

$$y_{ijc} \in \{0,1\} \qquad i,j\in V; c\in C \qquad (16)$$

$$L_{ic} \textit{ : real variables} \qquad i\in V; c\in C \qquad (17)$$

As stated, the objective function of the first step has a big impact on the solution space of the second step. Therefore, we implement successively two alternative objectives. Objective function z1 seeks to minimize the containers' total operating time. Conversely, objective function z2 solely seeks to minimize the travel time excluding the waiting and service times at the customer and terminal nodes:

$$\min z2 = \sum_{i,j\in V} \sum_{c\in C} y_{ijc} * t_{ij} \qquad (18)$$

It is assumed that both objectives provide a promising basis for the vehicles' routes since z1 and z2 represent two variants for the minimization of the containers' flows. Due to the interdependency of the transportation resource and the means of transport, the minimization of the containers' flows will consequently cause a minimization of the vehicles' total operating time.

Restrictions (2) and (3) state that every customer and terminal node is visited once by a container. The conditions for the start and end vertices of the different kinds of containers are considered by restrictions (4) to (10). Thereby, inbound full containers need to be moved from the terminal to the receivers. While inbound empty containers begin their path at the terminal and are transported to a shipper or the depot, restriction (6) states that additional empty containers originate from the depot. These three types of containers are not allowed to start their path from a different node stated by (7) and (8). Constraints (9) and (10) assure that a container ends its tour either at the depot or the outbound empty terminal. As stated by restriction (11) a container which is moved to a shipper/receiver node has to pass both time windows since in between times the container's filling/emptying process is performed by the customer's service personnel. The route and time continuity is stated by (12) and (13). Finally, restriction (14) and (15) assure that a container reaches a location in its defined time window. Hereby, we want to ensure that objective z1 represents the exact containers' total operating time. Therefore L_{ic} takes the value *0* if container *c* is not moved to node *i*.

The second step illustrating the vehicles' routes can be formulated as follows:

$$\min\ z3 = \sum\nolimits_{c\in C}(T_{e(0)c} - T_{s(0)c}) \tag{19}$$

s.t.

$$\sum\nolimits_{j\in V}\sum\nolimits_{k\in K} x_{ijk} = 1 \qquad i\in V_C \cup V_T \tag{20}$$

$$\sum\nolimits_{j\in V} x_{s(0)jk} = 1 \qquad k\in K \tag{21}$$

$$\sum\nolimits_{i\in V} x_{ie(0)k} = 1 \qquad k\in K \tag{22}$$

$$\sum\nolimits_{k\in K} x_{P_i D_i k} = 1 \qquad i\in\{s(0)\}\cup V_{R^o} \cup V_{S^o} \cup V_{T^I} \tag{23}$$

$$\sum\nolimits_{j\in V} x_{jik} - \sum\nolimits_{j\in V} x_{ijk} = 0 \qquad i\in V_C \cup V_T; k\in K \tag{24}$$

$$T_{jk} \geq T_{ik} + t_{ij} + s_i - M(1 - x_{ijk}) \qquad i, j\in V; k\in K \tag{25}$$

$$a_i \leq T_{ik} \leq b_i \qquad i\in V; k\in K \tag{26}$$

$$x_{ijk} \in \{0,1\} \qquad i, j\in V; k\in K \tag{27}$$

$$T_{ik}\ \textit{: real variables} \qquad i\in V; k\in K \tag{28}$$

The objective function z3 seeks to minimize the total operating time of the used vehicles. The most important restriction of the second step is given by equation (23). Thereby, the determined origins and destinations of the empty containers of the first step are used as the input data for the vehicles' routes. The remaining model formulation is mainly adopted from the PDPTW. Restriction (20) ensures that every node is visited exactly once. A vehicle has to start and end its tour at the depot stated by (21) and (22). Constraints (24) and (25) assure the time and route continuity during a vehicle's route. Finally, a node's time window has to be held by an operating vehicle stated by (26).

2.3 Simultaneous Approach

Following the simultaneous approach the two sub-problems of the sequential approach are solved within one single step:

$$\min\ z3 = \sum\nolimits_{c\in C}(T_{e(0)c} - T_{s(0)c}) \tag{19}$$

s.t. (2)-(15)

(20)-(22) and (24)-(25)

$$\sum\nolimits_{k\in K} x_{ijk} \geq y_{ijc} \qquad i\in V_{S^o} \cup V_{R^o} \cup V_{T^I}; j\in V; c\in C \tag{29}$$

$$\sum\nolimits_{k\in K} x_{ijk} \geq y_{ijc} \qquad i\in V_C \cup V_T; j\in V_{S^i} \cup V_{R^i} \cup V_{T^O} \cup \{e(0)\}; c\in C \tag{30}$$

$$T_{ik} = L_{ic} \qquad i\in V_C \cup V_T; k\in K; c\in C \tag{31}$$

$$x_{ijk}, y_{ijc} \in \{0,1\} \qquad i, j\in V; k\in K; c\in C \tag{32}$$

$$T_{ik}, L_{ic}\ \textit{: real variables} \qquad i\in V; k\in K; c\in C \tag{33}$$

By considering the containers' allocation on the one hand ((2)-(15)) and vehicle routing and scheduling (((19)-(21) and (23)-(24)) on the other hand, the presented model pursues the minimization of the vehicles' total travel time ((19)). Since a container cannot drive on its own, it has to be assured that every container is transported by a vehicle. Equations (29)-(31) require that the vehicles are interlinked with the containers and that both pass every location at the same time. Thereby, the vehicles cover the containers' routes but can skip the filling and emptying process of the container at a customer location.

3 COMPUTATIONAL RESULTS

To compare the 2-step approach with the integrated routing approach with respect to effectiveness and efficiency, we have generated ten test instances. Thereby, every single test instance is based on Solomon's benchmark R101-vehicle routing problem with time windows data set (Solomon (1987)). Since the defined MIP models are coded in CPLEX, only small data sets can be analyzed. Each instance comprises ten transportation requests.

Table 1. A comparison of the results for the 2-step approach

		2-Step-Approach Applying z1			2-Step-Approach Applying z2	
Instance	**Vehicles**	**Objective Value**	**Computation Time (in seconds; $1^{st}+2^{nd}$**	**Vehicle s**	**Objective Value**	**Computation Time (in seconds; $1^{st}+2^{nd}$**
1	6	600.07	10.52 (8.36+2.16)	5	621.27	4.01 (0.12+3.89)
2	6	825.96	8.17 (1.78+6.39)	4	915.12	1.55 (0.14+1.41)
3	6	671.46	19.89 (2.12+17.77)	4	717.20	12.15 (0.09+12.06)
4	5	767.45	5.10 (2.05+3.05)	5	862.50	3.62 (0.14+3.48)
5	8	996.90	3033.83 (2.03+3031.80)	5	1015.27	5.63 (0.19+5.44)
6	6	643.59	23.77 (1.77+22.00)	5	690.56	29.22 (0.14+29.08)
7	6	831.31	6.59 (1.95+4.64)	5	851.79	2.41 (0.14+2.27)
8	6	704.89	12.04 (2.09+9.95)	4	805.15	1.89 (0.09+1.80)
9	5	626.39	5.49 (3.63+1.86)	4	648.98	2.16 (0.11+2.05)
10	5	772.01	7.41 (2.30+5.11)	4	795.46	1.86 (0.14+1.72)

In detail, we randomly select four IF and four OF transportation requests as well as one IE and one OE transportation request. An aggravating factor according to the problem complexity is the consideration of two time windows at every customer location and the duplication of the terminal node for each transportation request. Hence, each test instance includes 28 nodes.

In the first instance we wanted to analyze whether objective z1 or z2 generates better solutions for the 2-step method. As it can be seen in Table 1, applying objective z2 dominates the application of z1 in terms of the number of operating vehicles. Thus, it can be concluded that applying z1 leads to the assignment of additional containers so that, in consequence, more vehicles are required to move them. Due to the fact that the employment of additional vehicles mostly induces a bigger solution space with more opportunities to solve the underlying problem instance, the first variant of the 2-step method leads to better objective values but also to worse results according to the computation time. Bearing in mind the global objective to minimize the total fulfillment costs of the operating company (i.e. the number of operating vehicles and the vehicles' total operating time) it can be stated that z2 is the dominating objective function for the first step of the sequential approach.

Subsequently, the results of the 2-step method are compared with those of the integrated routing-approach. Obviously, applying the holistic simultaneous approach always leads to the global optimum of the OD-CTTP and the generated results are, therefore, marked as benchmark values for the underlying problem type. Table 2 illustrates the results of both solution approaches. The assumption that the 2-step method generates a surplus of routes is only verified in instance 1 and 4 where one additional vehicle is required to serve all customers, respectively. Hence, for small test instances this hypothesis is scientifically not tenable if z2 is applied. Comparing the objective values, it can be concluded that the solutions of the sequential approach deviate on average 7% from the best solution. Needless to say that applying the 2-step approach has the big advantage of finding a solution much faster. The computational experiments show an extraordinary large advantage of 96% less computation time compared to the integrated routing approach.

Moreover, we made experiments to find out the limitations of the presented approaches in terms of the maximum problem sizes they can solve. The computation time needed to find a solution depends to a big extent on the characteristics of the instances. For the OD-CTTP the number of transportation requests and the time windows' wideness are the most affecting factors since they have a great impact on the operating of containers and vehicles which on his part will influence the computation time. Due to the fact that we implemented Solomon's R101-instances, the wideness for each time window is already fixed. Thus, testing the limitations of the manageable problem sizes only refers to a variation of the number of transportation requests, which we raised iteratively.

Table 2. Comparison of the Sequential and the Simultaneous Approach

	2-Step-Approach Applying z2			Integrated Routing-Approach		
Instance	**Vehicles**	**Objective Value**	**Computation Time (in seconds; 1st+2nd**	**Vehicles**	**Objective Value**	**Computation Time (in seconds)**
1	5	621.27	4.01 (0.12+3.89)	4	577.16	2964.73
2	4	915.12	1.55 (0.14+1.41)	4	809.44	39.12
3	4	717.20	12.15 (0.09+12.06)	4	676.55	126.12
4	5	862.50	3.62 (0.14+3.48)	4	745.84	215.83
5	5	1015.27	5.63 (0.19+5.44)	5	969.85	78.41
6	5	690.56	29.22 (0.14+29.08)	5	690.56	541.74
7	5	851.79	2.41 (0.14+2.27)	5	841.23	6601.06
8	4	805.15	1.89 (0.09+1.80)	4	685.74	67.56
9	4	648.98	2.16 (0.11+2.05)	4	572.18	100.34
10	4	795.46	1.86 (0.14+1.72)	4	795.46	49.00

In each iteration, we randomly tested three instances and stopped CPLEX's solving process after six hours if there still existed a gap to the lower bound. Considering the simultaneous solution approach, the limitation is reached if ten transportation requests have to be served. Conversely, CPLEX is able to solve instances with 19 transportation requests applying the 2-step method.

4 CONCLUSIONS

In this contribution, we have presented two methods to solve the OD-CTTP. Both methods consider not only vehicle routing and scheduling but also the allocation of the containers. While the sequential approach firstly solves an empty container repositioning problem and then uses the results as an input for the generation of the vehicles' routes on a second step, the simultaneous approach solves these two sub-problems within one single step. Computational experiments carried out on small instances show that the 2-step method reaches adequate results if the appropriate objective function for the containers' allocation is applied. Compared with the simultaneous approach the sequential approach led to results with the same number of required trucks in eight of ten instances by using 96% less computational time. Thus, for future work, it can be advisable to implement heuristic approaches for the 2-step method instead of simultaneous approaches. Thereby, efficient heuristics for the PDPTW known from the literature (e.g. Ropke and Pisinger (2006)) can be used for generating the vehicles' routes. Nevertheless, in doing so, further additional objective functions for the first step of the 2-step approach have to be developed and tested since, even for small test instances, the results for the vehicles' total operating time reached by the 2-step method deviate 7% from the global optimum.

REFERENCES

Kopfer, H., Sterzik, S. and Yun, W.Y.(2011), The potential of container sharing for trucking companies in seaport hinterland transportation, In C. Siepermann, M. Eley (eds), *Logistik gestern, heute, morgen (Festschrift fuer Richard Vahrenkamp)*, Germany, Berlin.

Parragh, S.N., Doerner, K.F. and Hartl, R.F. (2008), A survey on pickup and delivery problems - Part II: Transportation between pickup and delivery locations, *Journal fuer Betriebswirtschaft*, 2, 81-117.

Ropke, S. and Pisinger, D. (2006), An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transportation Science*, *4*, 455-472.

Solomon, M. M. (1987), Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, *2*, 254-265.

Toth, P. and Vigo, D., (2002). An overview of vehicle routing problems, In Toth, P., Vigo, D. (eds.), *The vehicle routing problem*, USA, Philadelphia.

Zhang, R., Wang, D. and Yun, W. Y. (2009), Model and algorithm of multi-depot container truck transportation with time windows, *Control and Decision Conference (CCDC)*, 148-153.

ANALYSIS OF VARIOUS DISPATCHING STRATEGIES AT SHORT SEA CONTAINER TERMINALS

Mustafa Egemen Taner[1], Osman Kulak[1], Olcay Polat[2] and Mehmet Ulaş Koyuncuoğlu[1]

[1]*Pamukkale University, Department of Industrial Engineering, Kinikli, 20100 Denizli, Turkey, metaner@pau.edu.tr, okulak@pau.edu.tr, ulas@pau.edu.tr*

[2]*Technical University of Berlin, Department of Production Management, Straße des 17. Juni 135, 10623 Berlin, Germany, opolat@pau.edu.tr*

ABSTRACT

Being advanced technologies, automated guided vehicles (AGVs) and automated lifting vehicles (ALVs) have been lately used in container terminals to improve efficiency as they are practical on the repetitive nature of the terminal operations. Due to their low berth deepness, container terminals in Turkey are built as artificially filled near the coasts. The most common layouts in these terminals are Π, *L*, or Ψ berth typed. In this paper, in order to analyze the effect of transporter number, transporter request rules and intersection rules on the performance of the short sea container terminals, an object-oriented simulation model is developed for berth type Π, *L* and Ψ. Actually, the effects of AGV dispatching rules on the determined performance criterion of total container handling in quay cranes is the issue we focus on. According to the results of the simulation, it can be said that terminal layouts have significant effects on the performance of the terminals and the number of the AGVs used.

Keywords: Short sea container terminals, AGV dispatching rules, object-oriented simulation

1 INTRODUCTION

Since the middle of the 20th century, more and more cargo such as furniture, toys, footwear, clothing, auto parts, electronics components, and computers is being containerized and export and import is increasing on a global scale. In today's just-in-time global supply chain, improving the efficiency of container shipping processes is more important than ever.

In seaports, container terminals are the places where container vessels are loaded and unloaded, and where the containerized cargo is temporarily stored while awaiting a future transportation.

These facilities can be classified as automated and non-automated considering the equipment technologies used. In automated container terminals, container info and automatic control technologies are used. These kinds of terminals have been established in Western Europe countries where work force is expensive. So, these terminals are more efficient than the others and also, the operation costs of these terminals are less than others. Moreover, in non-automated container terminals, the operations are carried out under the human control. In contrary, this kind of terminals has been established in South-East Asia countries where work force is cheaper (Steenken et. al., 2004).

In container terminals, three types of services such as import, export and transit are executed according to container trade types. One of these services is for import during which containers come by vessel and exit from gate; the other one is for export during which containers come by external transporters (XTs) and exit from berth by vessel; the last one is for transit during which containers come by vessel and exit by another vessel.

Different equipments are used while these services are being executed. In Fig. 1, services executed in a container terminal, transportation and handling operations and the equipments used during the operations are shown. Basically, three types of equipments are used in terminals. These equipments are; firstly quay cranes used for unloading/loading the container from/to vessel, secondly yard cranes used for stacking containers at yards for unloading/loading the container from/to transporter vehicle, and lastly the transporter vehicles used for transportation operations between berth and yard.

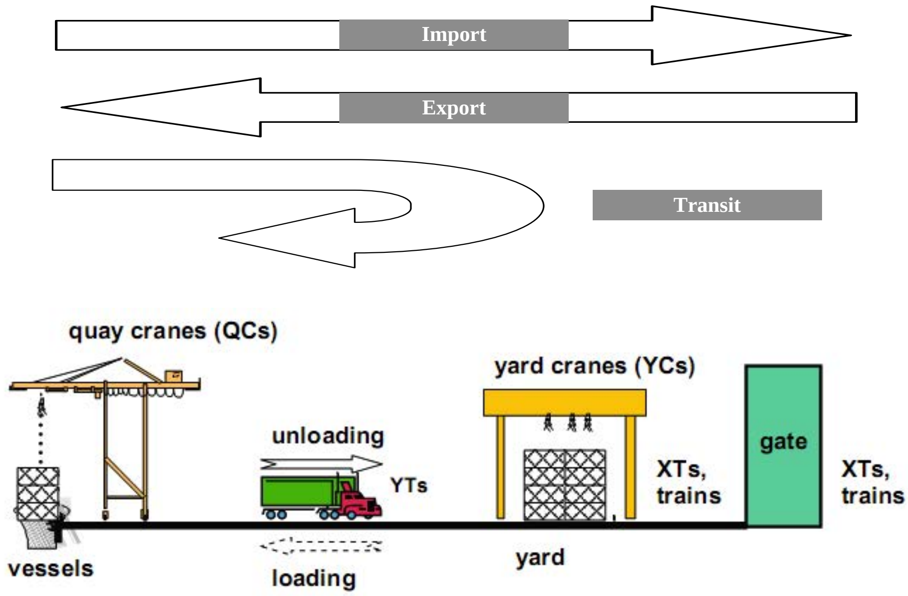


Figure 1. Terminal Services and Equipments (Petering and Murty, 2009)

In automated container terminals, AGVs are similar to conventional trucks, but they operate driverless on a pre-defined guide path. ALVs are vehicles which move over a container, lift it up and transport it to the designated storage location. Contrary to ALVs, AGVs need to interact with a crane. In recent years, Lift AGVs are the systems which can leave and take containers to the buffer areas without lifting them. In non-automated terminals, straddle carriers or conventional yard trucks (YTs) are used for moving containers inside the terminal. On the landside, many European container terminals have railway links which are not so common in most Asian countries (Kulak et. al., 2011).

In the literature, AGV dispatching rules were tested firstly by Egbelu and Tanchoco (1984). Additionally, Durrant-Whyte (1996) and Evers and Koppers (1996) are the first academic studies about AGV applications at container terminals. The design of AGV systems in container terminals is still a common problem. For comprehensive reviews of AGV systems in container terminal literature, we would like to refer to the surveys by Vis (2006) and Roodbergen and Vis (2009). Qiu et. al. (2002), Kim and Bae (2004), Bish et. al. (2005), Grunow et. al. (2006), Vis and Bakker (2008), Nguyen and Kim (2009) and Angeloudis and Bell (2010) are significant studies about AGV dispatching rules in automated container terminals.

Specifically, issues of logistics control in seaport container terminals have produced a wealth of publications in the scientific literature, cf. Stahlbock and Voß (2008) for a recent overview. Recently, there exist precious publications in the scientific literature especially about logistics control in seaport container terminals. Among the logistics control problems, the overall performance analysis of container terminals, the evaluation of alternative configurations, berth allocation, stowage planning, scheduling of the handling equipment, storage and stacking policies, quayside and landside transportation planning have been primarily examined. However, there is limited number of studies about layout analysis and layout effects on terminal performance.

In this study, AGV dispatching rules are tested for Π, *L* and Ψ berth typed container terminal layouts using simulation by the help of the performance criterion of the total container handling in quay cranes to demonstrate the effect of existing layouts on the performance of terminals under different transporter numbers, transporter request rules and intersection rules. Hence, an object-oriented simulation model has been developed by using Arena 10.0 Simulation Software. Some information about artificially filled type container terminals is given in the second section of the study. AGV dispatching rules are mentioned in the section 3 and the information about the designed container terminals is given in the section 4. In section 5, the simulation scenarios which have been developed for testing are mentioned and the results of these simulation scenarios are discussed with their statistically analysis. In the last part, the results of the study are presented and concluded.

2 ARTIFICIALLY FILLED TYPE CONTAINER TERMINALS

In the scientific literature, AGV applications in automated container terminals are used for the terminals in huge seaports. These huge seaports are all built in deep coasts, namely in natural ports. Normally, there exists only one major berth extending in parallel to the coast. Yet, yards are built perpendicularly or in parallel to the major berth in these terminals.

On the contrary, artificially ports, where there are shallow seas, may have several berths. Π, L, π or Ψ berth typed are the most common ones among these types. Berth and yard layout types in the artificially filled container terminals are mentioned below:

- Π berth type: The entire terminal is placed in the port which is artificially filled as peninsula. Yard area is placed in the middle of three berths which surround the peninsula. The terminals in Haydarpasa and Izmir ports are the examples for this type.
- L berth type: While the yard area and one berth of the terminal are placed in the port, the other berth is built by filling the sea. Sometimes, this filled berth can be used as partial yard area. The container terminal which is placed in Mersin port is an example of this type.
- Ψ berth type: While the yard area of the terminal is placed in the port, berths are placed in the long peninsulas which are filled perpendicularly to the sea. Sometimes, the berth which is artificially filled can be also used as partial yard area. Kumport container terminal in Ambarlı port is an example of this type.
- π berth type: While the yard area of the terminal is placed in the port, berths are placed in the long peninsulas which are filled horizontally to the sea. Sometimes, the berth which is artificially filled can be also used as partial yard area. Shangai ports in Yellow Sea are examples of this type.

It can be seen not only several areas in the terminal, but also horizontally or perpendicularly location of common yard area to the berths as a result of the fact that container terminals in these kinds of ports have several berths. It is pointed out in Vis and Anholt (2010), Polat et al. (2010) and Kulak et al. (2011) that the layout types effect the terminal performance in a significant way. In respect of the studies in the literature, due to this effect, AGV applications and dispatching rules in automated filled typed container terminals may differ.

3 AGV DISPATCHING RULES

In the frame of this study, AGV dispatching rules are analyzed using two different ways. These are: transporter request rules and intersection rules. Transporter request rules are used when a transporter is requested from the berth, yards, and gate. On the other hand, intersection rules used in order not to come across traffic jam or have an accident during AGVs' travel on the designed paths are shown in Fig. 2.

3.1 Transporter Request Rules

Smallest Distance: It is for selecting the available transporter nearest the requesting point.

Largest Distance: It is for selecting the available transporter farthest the requesting point.

Random: It is for selecting transporters randomly from the available transporter units.

Cyclical: It is for selecting the first available transporter unit beginning with the successor of the last unit selected.

3.2 Intersection Rules

FCFS – First Come First Served: The vehicle that arrived first at the end of its incoming link is given control of the intersection first.

LCFS – Last Come First Served: The vehicle that arrived last at the end of its incoming link is given control of the intersection first.

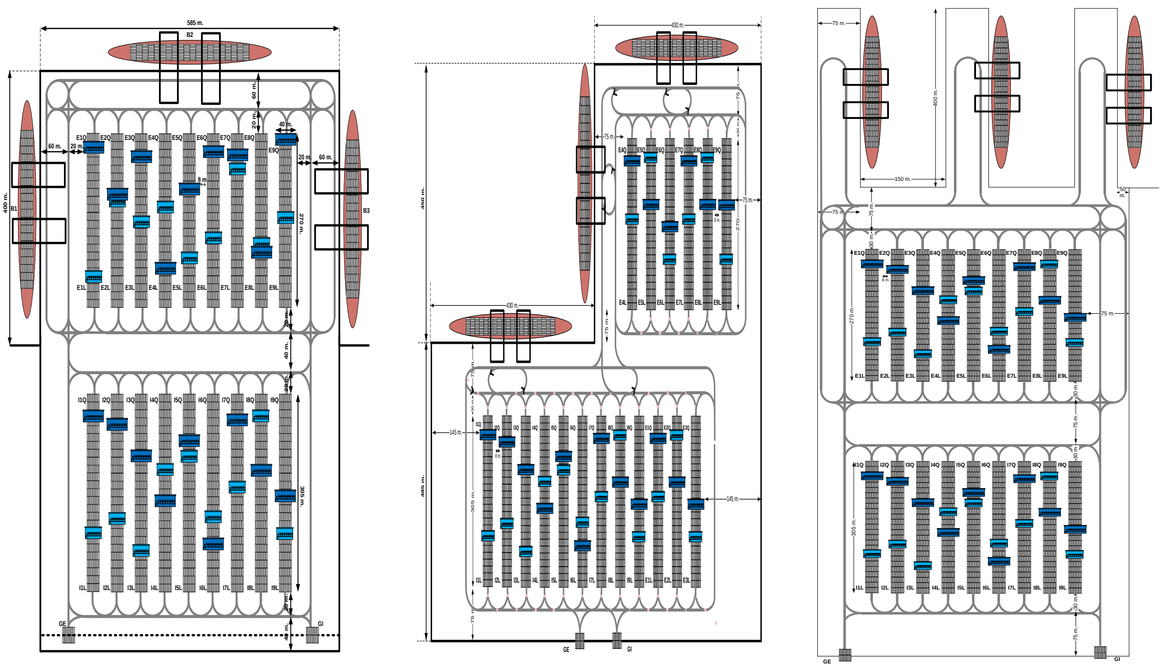
Closest: Giving the intersection to the vehicle closest to its travel destination.

HVF – High Value First: Giving the intersection to the vehicle whose controlling entity has the highest value of Attribute ID. The highest value means that the transporter is loaded with export container.

LVF – Low Value First: Giving the intersection to the vehicle whose controlling entity has the lowest value of Attribute ID. The lowest value means that the transporter is loaded with import container.

4 DESIGNED CONTAINER TERMINALS

In the structure of the study, large scaled Π, L and Ψ berth typed and also artificially filled container terminals which can be built in shallow coast areas have been designed. Layouts of these terminals which are designed as an automated container terminal are shown in Fig. 2 a, b and c respectively.



(a) Π berth typed **(b)** *L* berth typed **(c)** Ψ berth typed

Figure 2. Layouts of Designed Terminals

In each berth of these terminals, there are two dual automated quay cranes. Also, in the yard area there are 18 blocks (Π and Ψ berth typed – 9 blocks near to the gate for import containers, 9 blocks near to berths for export containers; L berth typed – 9 blocks near to the gate for import containers, 6 blocks near to berths for export containers and 3 blocks near to the other blocks). In each block, twin automated stacking cranes, which have telescopic design, are assigned. For export containers, each block length is 270 m. (44 TEU), width is 40 m. (10 TEU) and height is 13 m. (5 TEU). Additionally, for import containers, each block length is 305 m. (50 TEU), width is 40m. (10 TEU) and height is 13 m. (5 TEU). In this condition, simultaneous stack capacity of the each terminal is 42.300 TEU.

A simple presentation of AGV paths in the terminals are also shown in the figures above. AGV load/unload zones in blocks are located at the endpoints of each block. For export containers, container loading zone is located at the berth side and container unloading zone is located at the landside of block. For import containers, container unloading zone is located at the berth side and container loading zone is located at the landside.

In these container terminals, there are buffer areas which have the capacity of 20 containers at each block and 10 containers at each berth because of lift AGVs. Owing to these buffer areas, yard cranes and AGVs can run loading/unloading operations independently. Hence, it is possible to decrease waiting times of terminal equipments considerably. Optimum number of AGVs used in terminals has been obtained separately for each simulation scenario and for each berth typed terminal because of the differentiating number of AGVs according to the tested rules.

5 SIMULATION TESTS

5.1 Simulation Model

In the simulation model which has been developed by using Arena 10.0, vessels are the entities. Features of the vessels (loading/unloading amounts, vessel types – large, medium and small) are assigned and these vessels are allocated to berths in the first part of the model. In the second part, the operations executed at berths are mimicked. Loading / unloading processes of the containers in these vessels, and assignment of the containers unloaded from the vessel to the yards and AGV assignments are carried out. In the last part, yard area operations are simulated.

For the simulation tests, collected data are; inter arrival times of vessels, load to be charged onto a vessel, load to be discharged from a vessel, handling time of quay cranes, handling time of yard cranes, travel time of AGVs, average storage time in yard blocks (Import & Export), Import/Export ratio, equipment availability of quay cranes, equipment availability of yard cranes, equipment availability of AGVs. These data were analyzed using Arena Input Analyzer 10.0 and SPSS 16 following the concept of trace-driven simulation. Distributions with their parameters were determined with respect to minimum squared errors based on the Chi Square Test of the Arena Input Analyzer.

In this paper, we have some assumptions that normally ignore the minor problems but concentrate mostly on the major ones. Below are these major assumptions of the simulation model:

- YTs and AGVs can carry only one container
- Operating conditions of the terminal are not affected by weather conditions and they do not differ between the working shifts.
- We also assume that vessel arrivals at berths are unscheduled and thus, considered as random events and as a result of this, the collected data may differ.

In this context, the simulation tests are implemented under 3.000.000 TEU demand with 5 replications and one-year simulation time, and total container handling at QCs per year is used as the performance criterion.

5.2 Numerical Results for the Design of Π Berth Typed Terminal

In the first experiment set, the simulation model is used for analyzing transporter request rules in Π berth typed terminal. For this purpose, 4 different transporter request rules for AGVs are tested. In the scenarios, 'closest' is used as intersection rule. These scenarios are also tested with different transporter numbers in order to obtain the optimum transporter number. The effect of transporter request rules using different transporter numbers on total container handling at QCs per year is presented in Fig.3.

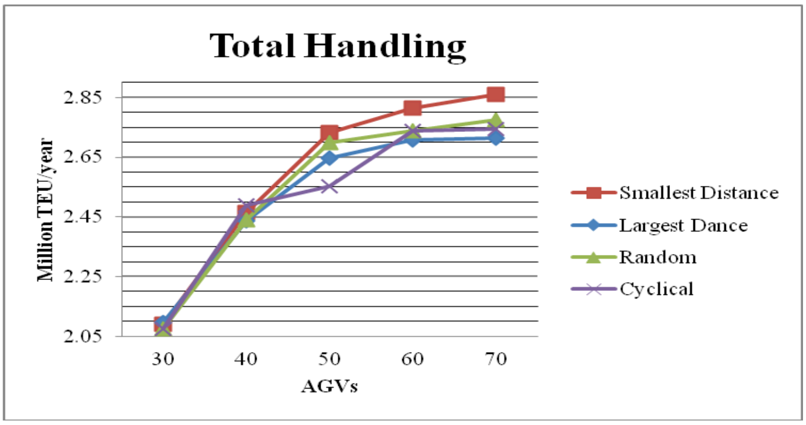


Figure 3. The Effect of Request Rules Under Different Transporter Numbers

As mentioned in Fig. 3, there is a significant difference between the request rules under different transporter numbers. In Table 1, the results of Tukey HSD tests which are carried out to obtain the differences above are in favor of which transporter request rule are presented. Simulation results which are related to 5 different transporter numbers are used in tests for transporter request rules.

Table 1. Tukey HSD Test for Transporter Request Rules

		Homogeneous subgroups (Alfa = 0,05)	
Transporter Request Rules	N	1	2
Cyclical	25	2520030,88	
Largest Distance	25	2520552,96	
Random	25	2545337,60	
Smallest Distance	25		2597987,20

According to Table 1, it is obvious that the transporter request rule which has the highest handling amount is the smallest distance. So, assigning AGVs to containers with smallest distance rule will provide higher handling amount as it provides time-saving and decreasing in waiting times. In Table 2, the results of Tukey HSD test which are used for obtaining the most appropriate transporter number under smallest distance rule are presented.

Table 2. Tukey HSD Test for Transporter Numbers

		Homogeneous subgroups (Alfa = 0,05)			
Transporter Numbers	N	1	2	3	4
30	5	2089241,60			
40	5		2463484,80		
50	5			2730870,40	
60	5			2815529,60	2815529,60
70	5				2858809,60

As mentioned in Table 2, it is clear that there is no notable difference between the numbers of 50 and 60 transporters and the numbers of 60 and 70. Therefore, it is more cost-effective use to analyze the ranges of 50-60. So, for 51-54-57-60 numbered transporters, Tukey HSD test is again carried out to find out the optimum number of transporters between the ranges of 50-60. In table 3, Tukey HSD test results are shown.

Table 3. Tukey Hsd Test for Transporter Numbers

		Homogeneous subgroups (Alfa = 0,05)	
Transporter Numbers	N	1	2
51	5	2737321,60	
54	5		2773305,60
57	5		2813152,00
60	5		2815529,60

In Table 3, there is a significant difference between the numbers of 51 and 54, 57, 60 transporters. However, there is no significant difference between the numbers of 54, 57 and 60 transporters. In this paper, although there is no feasibility study related to lift AGVs, the optimum number for these transporters are determined as 54 because of high purchase and installation.

In the second experiment set, the simulation model is used for evaluating the scenarios with defined intersection rules in terminals. For this reason, 5 different intersection ruled scenarios for the transporter number 54 and 4 different transporter request ruled scenarios are tested. In Fig. 4, the effect of intersection rules on the performance criterion under different transporter request rules is shown.

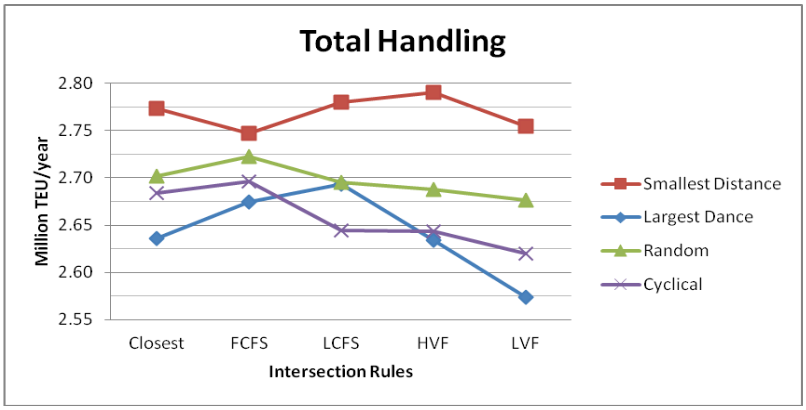


Figure 4. The Effect of Intersection Rules Under Different Transporter Request Rules

According to Fig. 4, transporter request rule and intersection rule which maximize total handling amount per year are selected for Tukey HSD test. Considering test results and Fig. 4, it can be said that the best transporter request rule is “smallest distance rule” among the other request rules. Smallest distance rule is different from the other transporter request rules in terms of being at %95 confidence level. In the same way, HVF is the best intersection rule at %95 confidence level among the other intersection rules.

The simulation tests which are applied on Π berth typed terminal are also implemented on the other berth typed terminals. Results of the tests according to the berth types are shown in Table 4.

Table 4. Results of Simulation Tests According to the Berth Types

Berth Type	**Optimum Number of AGVs**	**Best Transporter Request Rule**	**Best Intersection Rule**	**Total Handling under Best Rules (TEU)**
Π berth typed	54	Smallest Distance	High Value First (HVF)	2.789.965
***L* berth typed**	57	Cyclical	Last Come First Served (LCFS)	2.664.266
Ψ berth typed	96	Smallest Distance	Closest	2.159.527

6 CONCLUSION

In this paper, simulation models are developed to analyze transporter number, transporter request rules and intersection rules in Π, *L* and Ψ berth typed designed container terminals. Implemented tests show that the best scenario is; the scenario with using 54 lift-AGVs, "smallest distance rule" as request rule, "HVF" as intersection rule for Π berth typed terminal configuration, the scenario with using 57 lift-AGVs, "cyclical rule" as request rule, "LCFS" as intersection rule for L berth typed terminal configuration and the scenario with using 96 lift-AGVs, "smallest distance rule" as request rule, "Closest" as intersection rule for Ψ berth typed terminal configuration.

For Π berth typed terminal configuration, when the results are compared with Vis and Bakker (2006), smallest distance rule is familiar for transporter request rule in each study. However, in related study FCFS is proposed as intersection rule but, HVF is proposed in this study. The reason for this difference can be the different terminal berth types. In the Ψ berth typed terminal configuration, the amount of total handling under best rules is lower than the other types. The main reason for this is that the distances between yards and berths are different. These distances are particularly farther in Ψ berth typed than in Π and L berth typed. Even though it seems like a disadvantage, there are also some advantages of the Ψ berth typed terminal configuration such as their facility of being established at lower cost due to their amount of fill and of serving much more vessels with less space at the same time.

With the help of this study, we have identified that the transporter request and the intersection rules can behave differently according to terminal berth types. Consequently, the effect of these rules will be researched for the other artificial and natural berth types of container terminals. Furthermore, the effect of allocation strategies as a part of this research has also been studied on various types of container terminal layouts.

REFERENCES

Angeloudis, P. and Bell, M.G.H. (2010), An uncertainty-aware AGV assignment algorithm for automated container terminals, *Transportation Research Part E*, 46 (3): 354-366.

Bish, E.K., Chen, F.Y., Leong, Y.T., Nelson, B.L., Ng, W.C. and Simchi-Levi, D. (2005), Dispatching vehicles in a mega container terminal, *OR Spectrum*, 27:491–506.

Durrant-Whyte, H.F. (1996), An autonomous guided vehicle for cargo handling applications, *International Journal of Robotics Research*, 16: 407–440.

Egbelu, P.J. and Tanchoco, J.M.A. (1984), Characterization of automatic guided vehicle dispatching rules, *International Journal of Production Research*, 22: 359-374.

Evers, J.J.M. and Koppers, S.A.J. (1996), Automated guided vehicle traffic control at a container terminal, *Transportation Research A*, 30 (1): 21–34.

Grunow, M., Guenther, H-O., and Lehmann, M. (2006), Strategies for dispatching AGVs at automated seaport container terminals, *OR Spectrum*, 28:587–610.

Kim, K.H., and Bae, J.W. (2004), A look-ahead dispatching method for automated guided vehicles in automated port container terminals, *Transport Science*, 38(2):224–234.

Kulak, O., Polat, O., Gujjula, R. and Guenther, H-O. (2011), Strategies for improving a long-established terminal’s performance: A simulation study of a Turkish container terminal, *Flexible Services and Manufacturing Journal*, DOI: 10.1007/s10696-011-9128-x.

Nguyen, V. and Kim, K. (2009), A dispatching method for automated lifting vehicles in automated port container terminals, *Computers& Industrial Engineering*, 56:1002–1020.

Petering, M.E.H. and Murty, K.G. (2009), Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal, *Computers and Operations Research*, 36: 1711–1725.

Polat, O., Kulak, O., Taner, M.E. and Guenther H-O. (2010), Effect of Resource Allocation Rules in Different Layout Types of Seaport Container Terminals, *24th European Conference on Operational Research*, Lisbon, Portugal, July 2010.

Qiu, L., Hsu, W-J., Huang, S.Y. and Wang H. (2002), Scheduling and routing algorithms for AGVs: a survey, International Journal of Production Research, 40: 745–760.

Roodbergen, K.J. and Vis, I.F.A. (2009), A survey of literature on automated storage and retrieval systems, *European Journal of Operational Research*, 194, 343-362.

Stahlbock, R. and Voß, S. (2008), Operations research at container terminals: A literature update, *OR Spectrum*, 30: 1-52.

Steenken, D., Voß, S. and Stahlbock, R. (2004), Container Terminal Operation and Operations Research – A Classification and Literature Review, *OR Spectrum*, 26: 3-49.

Vis, I.F.A. (2006), A comparative analysis of storage and retrieval equipment at a container terminal, *International Journal of Production Economics*, 103: 680-693.

Vis, I.F.A. (2006), Survey of Research in the Design and Control of Automated Guided Vehicle Systems, *European Journal of Operational Research*, 170(3), 677-709.

Vis, I.F.A. and Bakker, M. (2008), Dispatching and layout rules at an automated container terminal, *Progress in material handling research: 2008, K Ellis et al. (eds)*, Material Handling Institute, Charlotte, North Carolina, 685-703.

Vis, I.F.A. and Van Anholt, R.G. (2010), Performance analysis of berth configurations at container terminals, *OR Spectrum*, 32(3), 453-476.

INVESTIGATING THE S&T ACTIVITIES IN THE GLOBAL VALUE CHAIN OF THE BULGARIAN AUTOMOTIVE ENTERPRISES

Victoria Gedinach[1] and Diana Antonova[1]

[1]*University of Ruse, Faculty of Business and Management, Bulgaria, vgedinach@uni-ruse.bg, dantonova@uni-ruse.bg*

ABSTRACT

In the context of globalization the division of manufacturing processes between different countries as a result of globalization of value chains becomes increasingly important. Cooperation in research and development (R&D) and innovation activities is a part of the globalization of value chains and is characterized by: (1) widening the range of scientific and technological (S&T) tasks, which are solved through joint efforts; (2) increasing the costs for R&D activities; (3) emergence of new collaborative R&D organizations; (4) growth of integration between large, medium and small industrial enterprises, combining their advantages; (5) active collaboration between universities and private sector; (6) increasing the share of government funding to support collaborative research, etc. In comparison with other European countries with developed innovation activity, Bulgarian foreign affiliates of multinational companies account for a smaller share of S&T activities than of the processing sector.

Keywords: S&T activities, global value chain, industrial enterprises, automotive industry

1 INTRODUCTION

An effective S&T activities management is very essential in terms of globalization of value chains when the manufacturing processes are divided between different countries. Joint programs in the S&T sphere amalgamate researchers' efforts and thus can lead to results that some countries are not able to achieve individually. One of the biggest consumers of new high technologies and investors in S&T activities is the automotive industry. Its current development is characterized by dynamic transfer of knowledge and technologies and high S&T activities costs. They are a key factor of economic growth, ensuring of employment and economic competitiveness increase. During the transition to a knowledge-based economy, the ability to create, implement and spread new knowledge gains the strategic importance in the production of new high-tech goods and science-absorbable services (Sturgeon T. et al (2009)). The problem that should be studied in this report consists in analyzing the management of S&T activities of the Bulgarian automotive enterprises it terms of globalization of production networks in the automotive industry. Therefore the following hypothesis is formulated:

H0: In the context of globalization Bulgarian enterprises offering motor vehicles, trailers and semi-trailers manufacturing (MVTSM) are characterized by relatively weak competitiveness, that is related to the low S&T costs (including R&D and design). Hence, the share of scientific and technological activities has an insignificant influence on the automotive industry value chain creation.

The aim of this paper is to analyze the state of S&T activities during the global value chain creation in Bulgarian industrial enterprises for motor vehicles, trailers and semi-trailers manufacturing. The following tasks are set: (1) to make an evaluation of the competitiveness of Bulgarian enterprises for MVTSM; (2) to calculate the index of the importance of MVTSM in the economy of Bulgaria (by analysis of the macro environment of these enterprises); (3) to determine the importance of S&T activities in the main activity of the enterprises for MVTSM (which can confirm or refute the hypothesis of the study).

2 EXPOSITION

Motor vehicles, trailers and semi-trailers manufacturing in Bulgaria is a relatively young sub-sector. The production capacity of automotive enterprises in the country is concentrated mainly in the manufacturing of parts and accessories for motor vehicles and engines. Over 90% of Bulgarian enterprises for motor vehicles, trailers and semi-trailers manufacturing are private property and operate for the domestic, as well as for foreign markets. The average wage in the sub-sector is about 18% lower than the average wage in manufacturing industry. Typical for the sub-sector is its flexibility and mobility – rapid reorientation to new market niches. The automotive sub-sector in Bulgaria consists mainly of small and medium-sized enterprises for orders of larger producers. After the order implementation they redirect to the production of other parts that are sometimes connected with other sub-sectors.

2.1 Evaluation of the competitiveness of Bulgarian enterprises for MVTSM

The sub-sector ”Motor vehicles, trailers and semi-trailers manufacturing” in Bulgaria has the following features: (1) covers about 1,6% of the industrial production of the country and creates 1,1% of the value added in industry; (2) the wage level in this sub-sector is average. The average monthly salary is 560 levs (about 287 euros) and employment – nearly 12 thousands (3,6% of all the employed in the processing industry); (3) the index of the industrial production and the turnover in the sub-sector in 2010 were improved by 48,8 and 36,4 points (table 1). At the same time, the producer prices were increased by 8, 5 percentage points.

Table 1. Production indices in sub-sector “Motor vehicles, trailers and semi-trailers manufacturing” in Bulgaria (in percentage points)

-	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2011	-4.2	-4.2	3.4	-	-	-	-	-	-	-	-	-
2010	6.2	2.9	4.0	1.6	-0.8	4.5	-4.6	1.1	4.6	0.0	0.8	1.1
2009	-1.3	-4.7	8.8	-3.0	9.1	-0.8	8.3	-1.1	5.2	5.9	3.5	3.8
2008	3.2	-1.5	-6.4	0.8	-6.9	0.2	-5.0	-7.9	0.9	-4.3	-8.5	-5.4
2007	-4.5	7.3	1.9	-0.7	4.1	-3.9	2.1	4.8	-1.5	-7.5	12.6	-10.5
2006	2.1	0.6	3.8	-0.2	1.2	5.3	-2.2	2.3	4.4	3.1	1.8	5.1
2005	4.3	2.9	4.5	4.6	5.2	0.9	6.3	-2.7	6.7	-0.1	4.5	-1.5
2004	6.0	-2.9	4.1	2.4	-1.9	3.8	2.6	-1.3	4.8	1.6	4.2	6.9
2003	-7.3	4.0	0.5	0.2	0.7	0.2	1.2	-5.8	7.6	1.2	-3.2	0.7
2002	4.6	-4.6	50.5	-29.7	-2.9	-4.7	7.6	5.7	1.6	2.5	177.4	-62.8
2001	-0.5	0.3	5.0	-9.9	3.3	0.5	-7.1	-5.3	0.4	-6.9	10.1	-7.5
2000	-	-5.9	0.3	6.9	-8.7	6.4	-0.7	-0.7	6.6	-7.4	8.4	-3.8

Source: NSI (National Statistical Institute)

The automotive sub-sector in Bulgaria comprises the production of motor vehicles and engines; bodies for motor vehicles; trailers and semi-trailers; electronic and electrical parts for motor vehicles; other parts and accessories for motor vehicles. It is performed by about 140 enterprises, nearly 40% of them are located in the South planning region, and with the South Central and Northeast planning regions the percentage exceeds 70. The reviving of the

MVTSM affects the export dynamics, which increased by 60,9% in 2010, while the import decreased by 5,3%. The participation of the sub-sector in foreign trade is estimated to 3,0% of the total Bulgarian export and 4,4% of its total import (Table 2). The provision of human resources in the sub-sector is the result of 17 universities, 103 vocational schools, one private college and five elementary schools functioning.

Table 2. Index of turnover and export in sub-sector "Motor vehicles, trailers and semi-trailers manufacturing" in Bulgaria (in percentage points)

Total turnover				**Turnover – internal market**				**Turnover - foreign market**		
2010	**March 2011**	**Juny 2011**	**2009**	**2010**	**March 2011**	**June 2011**	**2009**	**2010**	**March 2011**	**June 2011**
74,05	84,17	75,72	53,76	84,86	49,18	39,02	51,82	88,52	103,24	95,65

Source: NSI

In spite of the weak development of MVTSM in Bulgaria, the potential for its development is relatively high, in view of the proximity to markets with high growth potential and low production costs. As a disadvantage can be indicated the low productivity of the sub-sector in the country, but it is due to the use of outdated equipment and technologies. Significant direct foreign investments in the S&T activities are reported in the class "manufacturing of parts and accessories for motor vehicles and engines". The S&T activities in the automotive industrial sphere are carried out at the Technical University (Sofia), IFAG, City University and American University. There are two clusters in Bulgaria: Regional Industrial Cluster "Electromobile" (IKEM) and International "Autocluster" – Technical University of Gabrovo. In Table 3 SWOT-analysis of the automotive industry in Bulgaria is presented.

Table 3. SWOT- analysis of the MVTSM in Bulgaria

Strengths	**Weaknesses**
1. High quality of the outputs 2. High degree of scientific specialists creativity 3. Relatively high degree of patents' purity observing and new products' protection 4. Availability of own funds and equipment for carrying out S&T activities 5. New products/services/production processes developing in cooperation with foreign enterprises for MVTSM 6. High potential of scientific specialists 7. High level of leadership style application to enhance the creative potential of personnel	1. Low costs of internal organizational S&T activities from the enterprise's general budget 2. Low costs of the vocational education level raising 3. Small amount of state funding for the innovation activity of industrial enterprises 4. Relatively weak employees' knowledge and skills in the field of automotive industry 5. Low level of the team work principle implementation 6. Low quality of the activity of the authorities for scientific and technological information spreading
Opportunities	**Threats**
1. A thriving domestic component production industry, which should benefit greatly from expanding completely built unit (CBU) production throughout Central and Eastern Europe (CEE) 2. Close proximity to countries where CBU production is expanding, creating demand for components, particularly Romania, where Automobile Dacia has commenced production of Renault Logan 3. Opportunity to participate in high-tech development projects (international cooperation of the enterprises for MVTSM in the S&T sphere through the participation of Bulgaria in the global automobile cluster) 4. Establishment of a car plant by China's Great Wall 5. EU accession gives Bulgaria's automotive supply sector access to tap growth of neighbouring countries' assembly outputs 6. Long term prospects for the haulage sector and therefore commercial vehicle sales 7. Attracting foreign investors in the sub-sector	1. Rising unemployment threatens consumer spending 2. Any future policies restricting the supply of consumer credit would hamper sales growth 3. Liquidity still tight 4. Heavily reliant on imports 5. Used cars account for around 85% of the total vehicle stock, making it difficult for importers of new vehicles 6. One-third of new cars are bought on credit, raising questions about the sustainability of consumer demand 7. Lack of transparency and hitherto little effort to combat fraud and corruption 8. Commercial vehicle sales have fallen dramatically 9. Lack of special schools with specialties oriented to the automotive industry 10. Low state costs of S&T activities in the enterprises for MVTSM

Source: adapted from Business Monitor International Ltd (2009)

Bulgaria is one of the most attractive destinations in Central and Eastern Europe for investments in Manufacturing. The country has the opportunity to develop different forms of collaboration with major motor vehicle manufacturers. Such cooperation can help to solve common socio-economic missions (increasing incomes in the state budget through tax deductions, extra work places etc.) and to include the Bulgarian automotive industry in the global automotive value chain.

In Bulgaria there is no production and consumption of local automobiles, although the demand of automobiles is growing continuously. Experts in the field of automotive industry forecast that the local automobiles' production and realization can be carried out in a long period of time (in 10-12 years).

This study includes 42 Bulgarian industrial enterprises that specialize mainly in the finished products/processes manufacturing to developed schemes/manuals or instructions for the S&T activities management, given from global enterprises-suppliers. The research results show that in terms of the companies' ownership forms, 100% of them are private property (48% are the property of foreign companies, for example Belgium, England, Germany, Turkey etc.): 23 of the investigated organizations (55%) are subsidiaries of large auto companies. They are contractors and/or suppliers of large international companies that outsource a certain activity (outsourcing contract). According to the investigation, R&D activities are realized in 14 of the surveyed enterprises (33%), 16 enterprises (38%) realize R&D activities in cooperation with parent companies, and the remaining 12 enterprises (29%) do not currently perform such activities. The reasons are high STA costs and/or lack of highly qualified staff in the field of automotive industry (it is important to note that the latter enterprises used to have R&D and engineering departments in the past).

2.2 Role of the MVTSM in the economy of Bulgaria

The adapted methodology of N. Konahina (2008) applied to the task of calculating the index of the class "Manufacture of motor vehicles" importance in the economy of Bulgaria allows the comparison of the MTVTSM importance in Bulgaria with the importance of the sub-sector in other countries. This comparison may be done by 2005, since there are not complete databases for the period after 2006 for certain countries. But the data used for 2005 allows us to calculate the necessary indicators for evaluating the condition of the automotive sub-sector and the calculating of other indicators that are applied to the methodology will balance the inability of the current index calculating.

The importance of the automotive industry, the prospects of its development and its overall role in the economy of the country are determined by the position the automotive transport in the transport and energy infrastructure. On the basis of the methodology used for calculating the index of the national/domestic motor vehicles, trailers and semi-trailers manufacturing in the national economy was calculated the index of the MVTSM importance in Bulgaria. This index (R) should be evaluated first in terms of production and second – the consumption of the products for the automotive industry. Variables from the first group are: state/national incomes from the sub-sector (Pr); employees in the MVTSM (Emp); investments in the sub-sector (I); share of production from related (drel) and supported branches (dsup) in automotive industry. The second group of variables can include: length of the road network of the country (Rn); share of automobile transport in the load-passenger transport (dal,dap); share of production with civil (dv1), military (dv2) and production (dv3) function.

$$R = f\,(Pr, Emp, I, Rn, Drel, Dsup, Dag, Dap, dv1, dv2, dv3) \qquad (1)$$

In terms of production the automotive industry (through the interaction with micro and macro environment) contributes to the expansion of tax base and the increase of revenues in the state budget; provides extra work places in the economy of the country; encourages the development of related and supported industries, investment, financial-credit institutions. The role of the automotive industry based on the consumption is related to: (1) national transport complex; (2) development of its infrastructure and (3) ensuring the independence and defensive capacity of the country.

Table 4. Variables that determine the role of MVTSM in the economy of the country

External micro and macro environment of the automotive industry	Variables	Stage of reproduction	Role of the industry in national economy
1. State (Government)	State incomes from the sub-sector (Pr)	production	Increasing the revenues in the state budget
2. Institutions – employment agencies	Employees in the sub-sector (Emp)	production	Providing extra work places in the economy of the country
3. Investment and financial-credit institutions	Investments in the MVTSM (*I*)	production	Development of investment and financial-credit institutions
4. Infrastructure	Length of the road network of the country (Rn)	consumption	Development of infrastructure of the country
5. Load-passenger transport	Share of automobile transport in the load-passenger transport (dal,dap)	consumption	Development of the transport sub-sector
6. Related and supported branches	Share of production from related (drel) and supported branches (dsup) used in automotive industry	production	Development of related and supported industries
7. Production with civil, military and production function	Share of production with civil (dv1), military (dv2) and production (dv3) function	consumption	Ensuring the economic independence and defensive capacity of the country

Source: N. Konahina (2008)

The calculation of the index of the automotive industry's importance in the economy of Bulgaria has probabilistic character, since the supposed importance of the above variables is applied by setting their weighting factors in equation 1. The weighting factors (k) for the variables that determine the role of the MVTSM in Bulgaria are as follows: for Pr k=0,3; for I – 0,3*;* for Emp-0,3; for Drel and Dsup – 0,1; for Rn – 0,2; for dal and dap – 0,1*;* for dv1 and dv3 – 0,5; for dv2 – 0,2. The variables' values for 2005 are presented in Table 4:

Table 5. Values of variables for calculating the index of MVTSM importance in the economy of Bulgaria (2005)

Variables	Values from 2005
1. Pr – state revenues from MVTSM (relative share from the general state incomes)	2,385,907 euro (0,02%)
2. Emp – employment in the sub-sector (relative share from the general employees in the industry)	3,329 people (0,10%)
3. *I* - investments in MVTSM (relative share from the general size of investments of Bulgaria)	26,500,000 euro (0,27%)
4. Relative share from the production of related (Drel) and supported (Dsup) industries	47,94%
5. Rn – automobile road network (relative share from the total state area)	38000 km (11,4%)
6. Relative share of automobile transport in the load-passenger transport (Dag,Dap)	65%
7. Relative share of production with civil (dv1), military (dv2) and production (dv3) function	2,869,443 euro (0,036%) 215,979 euro (0,003%)

Source: adapted to N. Konahina (2008)

Regarding production the index of the MVTSM importance in Bulgaria (R_1) for 2005 is as follows:

$$R_1 = f\,(Pr, Emp, I, drel, dsup) \qquad (2)$$

$$R_1 = f\,(0{,}00023; 0{,}001; 0{,}0027; 0{,}48) \qquad (3)$$

Entering the conventional weighting factors, the equation assumes the following view:

$$R_1 = (0{,}00023x0{,}3) + (0{,}001x0{,}3) + (0{,}00027x0{,}3) + (0{,}48x0{,}1) = 0{,}049 \qquad (4)$$

Regarding consumption the index of the MVTSM importance in Bulgarian economy (R_2) for the same year can be represented as:

$$R_2 = f\,(Rn, dal, dap, dv1, dv2, dv3) \qquad (5)$$

$$R_2 = f\,(0{,}114; 0{,}65; 0{,}036; 0{,}003) \qquad (6)$$

By the introduction of the conventional weighting factors the index can be defined as follows:

$$R_2 = (0{,}114x0{,}2) + (0{,}65x0{,}1) + (0{,}036x0{,}5) + (0{,}003x0{,}2) = 0{,}11654 \qquad (7)$$

Hence, R = 0,08.

The results of the calculations show the index of the MVTSM role in the economy of Bulgaria for 2005 has a relatively low value compared to other countries. This is due to the relatively weak sub-sector development (Table 5).

Table 6. Index of the national MVTSM importance in the economy of the country (by 2005)

National MVTSM	Index of the MVTSM importance	National MVTSM	Index of the MVTSM importance
1. USA	0,84	12. Czech Republic	0,25
2. Japan	0,53	13. Mexico	0,17
3. Germany	0,52	14. Korea	0,14
4. France	0,42	15. India	0,13
5. Spain	0,40	16. Poland	0,11
6. Italy	0,40	17. Brazil	0,09
7. United Kingdom	0,39	18. Russia	0,09
8. China	0,34	19. Bulgaria	0,08
9. Turkey	0,32	20. Thailand	0,02
10. Canada	0,27	21. Iran	0,00
11. Belgium	0,27		

Source: adapted from N. Konahina (2008)

In addition, the indices of the MVTSM importance in the economy of Bulgaria for 2006-2010 are calculated (the forecast index is calculated on the basis of pre-announced official statistics for 2010). Figure 1 shows the development of the MVTSM in Bulgaria between 2005-2010, based on these indices. The index of the MVTSM importance in the economy of Bulgaria for 2010 is approximately 0,09, which shows that it is still relatively low compared to that of other countries (even for 2005).

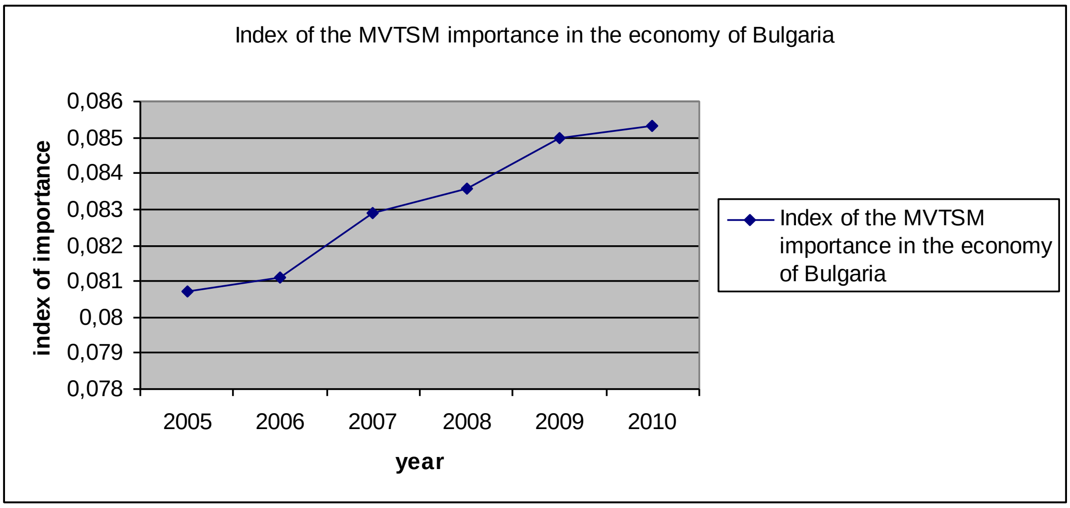


Figure 1. Development of MVTSM in Bulgaria on the basis of the index of the sub-sector's importance in the national economy (2005-2010)

2.3 Share of S&T Activities in the Bulgarian Enterprises for MVTSM

The automotive industry should be an industry, based on knowledge. The world automobiles, trailers and semi-trailers manufacturers spare out of about 4-6% of total sales volume for the S&T activities (Sturgeon et al, 2009). In terms of globalization of S&T the competitiveness of countries with MVTSM is determined by: (1) the scale of the S&T activities; (2) the technical and economic level and the production costs; (3) the S&T productivity; (4) the FDI flow and (5) international strategic cooperation etc.

On the basis of the study of the World Economic Forum (Global Competitiveness 2010), which covers 139 economies of developed and developing countries and aims to identify the competitive advantages of these countries the ranks of the competitiveness of particular countries on the basis of the S&T indicators are presented (Schwab, 2011). In Table 6 the main indicators of the S&T activities of Bulgaria for 2010 are analyzed.

Table 7. Ranks of the global competitiveness 2010-2011 (innovation indicators)

Indicator / Country	1. Capacity for innovation	2. Quality of scientific research institutions	3. Company spending on R&D	4. University-industry collaboration in R&D	5. Gov't procurement of advanced tech products	6. Availability of scientists and engineers	7. Utility patents per million population*
Developing countries							
CEE							
Turkey	55	89	62	82	62	**44**	70
Romania	72	83	103	103	105	55	62
Bulgaria	79	73	96	110	87	77	**31**

Source: K. Schwab (2011)

** Ranks of notable competitive advantages are highlighted. An asterisk (*) indicates that data are from the European Commission Joint Research Centre.*

The relatively poor development of the S&T activities in the Bulgarian enterprises for MVTSM is due to insignificant cooperation between S&T institutes and industrial automotive enterprises and relatively low costs for S&T activities at the enterprise level. The presence of a significant number of patents per million population is explained by relatively high quality of scientific and research institutions in Bulgaria.

The importance of S&T activities in these enterprises is determined through two criteria: (1) share of the costs for improving the vocational education level of the staff and (2) share of the S&T activities costs. The results of the study are analyzed in Table 7 and Table 8.

Table 8. Relative share of the costs for improving the vocational education level of the staff

Relative share of the costs for improving the vocational education level of the staff	**For qualification**		**For training**	
	Number	**%**	**Number**	**%**
below 0,5%	11	26,19%	12	28,57%
0,5%-1%	11	26,19%	10	23,81%
1%-2%	5	11,90%	5	11,90%
2%-3%	7	16,67%	7	16,67%
3%-5%	5	11,90%	5	11,90%
6%-8%	1	2,38%	1	2,38%
over 8%	1	2,38%	1	2,38%
Missing	1	2,38%	1	2,38%

Source: own elaboration

Table 9. Relative share of the S&T activities costs

Relative share of the S&T activities costs	**below 0,5%**	**0,5%-1%**	**1%-2%**	**2%-3%**	**3%-5%**	**6%-8%**	**over 8%**
Number	12	5	8	5	6	3	3
%	28,57%	11,90%	19,05%	11,90%	14,29%	7,14%	7,14%

Source: own elaboration

One of the most important aspects in the management of S&T activities in the context of globalization of value chains is to carry out joint S&T activities with different organizations (universities, R&D institutions, industrial houses, laboratories for quality control, research parks, business incubators etc.) with the purpose to ensure an effective relation „scientific environment – business environment". This study shows that 71,43% of investigated enterprises for MVTSM in Bulgaria have partners in the S&T activities implementation. The results of the study show that only 8 from 42 enterprises have certificates for patented inventions (19%). This is a comparatively good indicator, given the high level of the S&T dependence of the majority of surveyed enterprises on the contracting organizations.

As a result of this analysis it can be concluded that relatively important factors for increasing the competitiveness of Bulgarian enterprises for MVTSM are: (1) own S&T activities implementation and purchase of licenses for the foreign automobile models production; (2) increasing number of investment projects in cooperation with global automobile manufacturers. Thus the development of motor vehicles, trailers and semi-trailers manufacturing in Bulgaria should be subject to the following objectives: (1) providing the financial support to enterprises for MVTSM through the EU structural funds; (2) government funding of organizations to participate in international exhibitions and fairs with new automobile inventions; (3) improving the professional and educational qualification level (modernization of the programs for the professions in the sub-sector); (4) encouraging the cooperation between universities and automobile companies; (5) creation and support to clustering initiative

3 CONCLUSION

1. The Bulgarian participation in the global automotive value chain is justified by: (1) implementation of parts and accessories manufacturing for motor vehicles in cooperation with foreign automotive organizations; (2) participation in the global automotive cluster "Autocluster" (based on the analysis of the MVTSM enterprises' macro environment). Bulgaria's place in the global automotive value chain is expressed in subcontracting activities and its role as a subcontractor of the automotive production.

2. It is proved that in the context of globalization Bulgarian enterprises offering motor vehicles, trailers and semi-trailers manufacturing are characterized by relatively weak competitiveness, which is related to the low S&T costs (including R&D and design). Hence, the share of scientific and technological activities insignificantly influences the automotive industry value chain creation (confirmation of *H0*).

3. The index of the MVTSM importance in the economy of Bulgaria for the period 2005-2010 is calculated as a result of applied methodology for determining the role of the automotive industry in the economy of the country. This index allows to compare the automotive industry development in different countries and to fix its place in the economy of Bulgaria. The index of the MVTSM importance in the economy of Bulgaria is relatively small due to comparatively low values of the following indicators: (1) employment in the sub-sector; (2) state incomes from the national automotive industry; (3) investments in MVTSM.

4. The factors for increasing the competitiveness of Bulgarian enterprises for MVTSM are: own S&T activities implementation; purchasing foreign licenses for access to new technologies and increasing the number of investment projects in cooperation with global automobile manufacturers.

5. It is substantiated that the problem of the own S&T activities implementation in the Bulgarian enterprises for MVTSM is a consequence of the restrictions in the outsourcing contract, namely: subcontractors carry out the manufacturing process according to the set specifications of the contractor. The S&T activities are implemented by the organization-contractor, therefore the enterprise-subcontractor is unable to take part in the S&T activities management.

REFERENCES

ACEA (European Automobile Manufacturers Association), European automobile industry report 09/10, Brussels, 2009.

Business Monitor International Ltd, Bulgaria Autos Report Q1 2010, Mermaid House, London, 2009.

Dr. Christine Legner, Dino Pelli, Jan Löhe, European Business School; Jörg Walden, Theodor Fischer, Oliver Stein, iPoint-systems gmbh, Evolving Value Chains in the Automotive Industry − Implications for Business Processes and Information Systems, August 2009.

Eurostat, Europe in Figures, Eurostat Yearbook, 2009.

Konahina, N., The development of the global automotive industry in the context of globalization, synopsis of the dissertation, St. Petersburg, Russia, *pp. 4-15.*, 2008.

National statistical institute (2011), Bulgaria.

Organisation Internationale des Constructeurs d'Atomobiles, www.oica.net

Programme co-funded by the European Union, "Autoclusters Automotive Network for Innovations. Innovation Trends and Challenges and Cooperation Possibilities with R&D in Automotive Industry". South East Europe Journal for our common future,2010.

Schwab, K., The Global Competitive Report 2011-2012, *World Economic Forum*, Genewa, Switzerland, 2011.

Sturgeon, T., O., Memedovic, J., Biesebroeck, G., Gereffi (2009), Globalisation of the automotive industry: main features and trends, Int. J. technological learning, innovation and development, vol. 2, nos. ½.

Wyman, O., "Car Innovation 2015: A Comprehensive Study in Innovation in the Automotive Industry", MMC, 2007.

A CLUSTER-BASED HEURISTIC FOR ALLOCATING PRODUCTS IN MULTI-LEVELS WAREHOUSES

F. Guerriero[1], F. Mari[1], R. Musmanno[1], O. Pisacane[1] and F. Rende[1]

[1]*Dipartimento di Informatica, Elettronica e Sistemistica, Università della Calabria, Via P. Bucci 41C, Arcavacata di Rende, Italy*

ABSTRACT

One of the major issues in *Warehouse Management* is to optimally assign the product classes to the storage locations (*slots*, for short) on the basic principle that the most required items have to be allocated closer to the I/O doors (*Products Allocation Problem*-PAP). The aim of this paper is to study a special version of PAP considering a multi-layers warehouse with compatibility constraints among the classes (two aspects that, at the best of our knowledge, have not been addressed in scientific literature yet). First, we modelled the problem (as already described in Guerriero et al (2012)) with the aim of minimizing the total logistics costs (due to the handling operations and the products decentralization in the warehouse) satisfying specific operational constraints (for example, compatibility and capacity constraints). However, since on large-scale instances the complexity of the model (in terms of number of decision variables and constraints) becomes computationally intractable by optimization solvers, we also design, implement and test a cluster-based heuristic approach for overcoming this limitation. Finally, we compare the results from two points of views: the solutions quality and the computational overhead.

Keywords: Warehouse Management, Product Allocation Problem, Logistics, Graph Theory

1 INTRODUCTION

In the optimal management of warehouse operations, in terms of time and location, two main issues need to be addressed, that is the inventory management and the optimal products allocation.

The first one is related to the optimization of the storage costs, by guaranteeing a satisfactory service level; the second one is mainly focused on the reduction of the time required by the handling products operations. In this work, we focus our attention on the latter aspect, that is the *Products Allocation Problem* (PAP for short). The main aim is to assign the products to the warehouse's storage locations, in such a way that the handling cost/time is minimized, the space utilization is maximized and a set of operational constraints are satisfied.

Even though the strategy used to allocate the products in a warehouse strongly influences the warehouse performances, limited attention has been given in the scientific literature to the formulation and solution of the related problem (especially in presence of realistic constraints). Among the few published studies, we cite the work of Larson et al. (1997), which describes a three-phase heuristic, in which the last phase concerns the solution of the PAP; the paper of Muppani and Adil (2005), where a mathematical model to represent the products' aggregation and allocation problem is proposed; the work of Onout et al. (2008) that describes a mathematical model to address the layout and the PAP in a multi-layer warehouse and develops a particle swarm optimization meta-heuristic for its solution; the contribution of Heragu (2005), which addresses jointly the product allocation and layout design problems and proposes a heuristic approach to solve the formulated model; the recent paper of Sanei et al (2011), that presents a mathematical formulation of the PAP, taking into consideration various operational constraints, and a heuristic algorithm, based on the branch and bound method, to solve it.

Starting from this last work, some of the authors of the present paper have proposed in Guerriero et al (2012) a mathematical formulation of the PAP, that is based on the same allocation priorities described in Sanei et al (2011), but it also takes into account compatibility constraints among the products' classes and a multi-layers warehouse layout, aspects not addressed in Sanei et al (2011).

Since the mathematical model presented in Guerriero et al (2012) for real instances cannot be optimally solved, with limited computing resources and within a reasonable amount of computational time, in this work we develop a heuristic approach to address the PAP.

The main idea of the proposed strategy is to group the products into clusters, according to the compatibility constraints, and to successively allocate the compatible clusters to the storage locations.

The remainder of the paper is organised as follows. In Section 2, we outline the main characteristics of the problem and we report the mathematical formulation proposed in Guerriero et al (2012). Section 3 is devoted to a detailed description of the proposed cluster-based heuristic to solve the PAP. The proposed heuristic is evaluated on the basis of an extensive computational experimentation: the related results are presented and discussed in Section 4. The paper closes with some concluding remarks in Section 5.

2 THE MULTI-LEVELS PAP WITH COMPATIBILITY CONSTRAINTS

This Section illustrates and describes in details the general framework of the work that can be considered an extension of the one presented in Sanei et al (2011) to the case of a multi-layer warehouse and in presence of compatibility constraints among the products classes. It is worth noting that these last operational constraints have guaranteed to study a substantially different version of the problem from the one presented in Sanei et al (2011).

The general warehouse layout is characterized by 2 I/O doors (1 input and 1 output door); 2 main aisles (MA) and 2 transversal aisles (TA). Moreover, it is imposed that an even number of horizontal slots is present. The warehouse has also slots along the third dimension (height) according to the multi-layer scenario. A classes based assignment policy is used. Each class of products is characterized by a known demand (expressed in load units). The load units have the same dimension and the capacity is equal for each slot (expressed in load units too). According to the compatibility and capacity constraints, in each slot more than one classes could be allocated.

In the following: N $(1,\ldots,n)$ is the set of products classes; m and w are the number of vertical and horizontal slots, respectively; h is the number of levels; S is the set of the slots belonging to the first layer; C is a NxN binary matrix, whose component c_{ij} is 1 if class i is compatible to class j; P is an array of 5 possible discounts/penalties, where p_1, p_2, p_3 and p_4 are discounts applied if all the units of a class are assigned to the same block, to adjacent slots in the same aisle, to opposite slots and to posterior slots, respectively; p_5 is a penalty paid if a class is located in more than one slot.

Moreover, K is the set of the I/O doors; D, the distance matrix, where d_{slk} is the distance (in meters) of the slot in block s and layer l (for short (s,l)) from the door k; F, the handling matrix, where f_{ik} is the number of daily handling operations (in load units) of the class i from the door k. For each class i, r_i is the number of load units and, for each slot, Cap is the capacity (in load units).

Finally, T_1, T_2 and T_3 are the sets of the slots of the first level having one adjacent slot in the same aisle; an opposite; a posterior, respectively, and finally ε represents the cost for moving one load unit for a meter.

Definition 1
An instance of the PAP is feasible if and only if it satisfies the following condition: the total warehouse capacity W has to be greater or equal to the total demand (in terms of load units):

$$W \geq \sum_{i=1}^{n} r_i$$

The decision variables are in what follows: y_{isl}, the number of load units of class i assigned to (s,l) ($i = 1,\dots, n$, $s = 1,\dots, |S|$ and $l = 1,\dots,h$); x_{isl}, a binary variable equal to 1 if class i is assigned to (s,l) ($i = 1,\dots, n$, $s = 1,\dots, |S|$ and $l = 1,\dots,h$); z^1_{isl}, z^2_{isl}, z^3_{isl} and z^4_{isl}, binary variables equal to 1 if class i is allocated to the adjacent slot of (s, l) in the same block; to the adjacent slot of (s,l) in the same aisle; to the opposite slot of (s,l) and, finally, to the posterior slot of (s,l) in the same block, respectively ($i = 1,\dots, n$, $s = 1,\dots, |S|$ and $l = 1,\dots, h$); b_{is}, a binary variable equal to 1 if the class i is allocated to a slot of block s ($i = 1,\dots, n$ and $s = 1,\dots, |S|$)

2.1 A mathematical linear formulation

In this Section, by using the notation introduced in the Section 2, a mathematical linear formulation for the problem in exam is presented, as already proposed in Guerriero et al (2012).

$$\min \quad \varepsilon \sum_{k=1}^{|K|}\sum_{s=1}^{|S|}\sum_{l=1}^{h} d_{slk} \sum_{i=1}^{n} f_{ik}\frac{y_{isl}}{r_i} + \sum_{i=1}^{n}\left[\left(\sum_{s=1}^{|S|}\sum_{l=1}^{h} x_{isl}\right) - 1\right] p_5 - \sum_{i=1}^{n}\sum_{s=1}^{|S|}\sum_{l=1}^{h}(p_1 z^1_{isl} + p_2 z^2_{isl} + p_3 z^3_{isl} + p_4 z^4_{isl}) \quad (1)$$

s.a.

$$\sum_{s=1}^{|S|}\sum_{l=1}^{h} y_{isl} = r_i \quad , \forall\, i = 1,\dots,n \qquad (2)$$

$$z^1_{isl} \leq x_{isl} \quad , \forall\, i = 1,\dots,n,\ s = 1,\dots,|S|,\ l = 1,\dots,h-1 \qquad (3)$$

$$z^1_{isl} \leq x_{is(l+1)} \quad , \forall\, i = 1,\dots,n,\ s = 1,\dots,|S|,\ l = 1,\dots,h-1 \qquad (4)$$

$$z^1_{isl} = 0 \quad , \forall\, i = 1,\dots,n,\ s = 1,\dots,|S|,\ l = h \qquad (5)$$

$$z^2_{isl} \leq x_{isl} \quad , \forall\, i = 1,\dots,n,\ s \in T_1,\ l = 1,\dots,h \qquad (6)$$

$$z^2_{isl} \leq x_{i(s+1)l} \quad , \forall\, i = 1,\dots,n,\ s \in T_1,\ l = 1,\dots,h \qquad (7)$$

$$z^2_{isl} = 0 \quad , \forall\, i = 1,\dots,n,\ s \notin T_1,\ l = 1,\dots,h \qquad (8)$$

$$z^3_{isl} \leq x_{isl} \quad , \forall\, i = 1,\dots,n,\ s \in T_2,\ l = 1,\dots,h \qquad (9)$$

$$z^3_{isl} \leq x_{i(s+m)l} \quad , \forall\, i = 1,\dots,n,\ s \in T_2,\ l = 1,\dots,h \qquad (10)$$

$$z^3_{isl} = 0 \quad , \forall\, i = 1,\dots,n,\ s \notin T_2,\ l = 1,\dots,h \qquad (11)$$

$$z_{isl}^4 \leq x_{isl} \ , \forall\, i=1,\dots,n,\ s \in T_3,\ l=1,\dots,h \tag{12}$$
$$z_{isl}^4 \leq x_{i(s+m)l} \ , \forall\, i=1,\dots,n,\ s \in T_3,\ l=1,\dots,h \tag{13}$$
$$z_{isl}^4 = 0 \ , \forall\, i=1,\dots,n,\ s \notin T_3,\ l=1,\dots,h \tag{14}$$

$$b_{is} + b_{js} \leq 1 + c_{ij} \ , \forall\, i=1,\dots,n-1,\ j=i+1,\dots,n,\ s=1,\dots,|S| \tag{15}$$
$$b_{is} + b_{j(s+1)} \leq 1 + c_{ij} \ , \forall\, i=1,\dots,n,\ j=1,\dots,n, j \neq i,\ s \in T_1 \tag{16}$$

$$\sum_{i=1}^{n} y_{isl} \leq Cap \ , \forall\ s=1,\dots,|S|,\ l=1,\dots,h \tag{17}$$

$$y_{isl} \leq M x_{isl} \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S|,\ l=1,\dots,h \tag{18}$$
$$b_{is} \leq \sum_{l=1}^{h} x_{isl} \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S| \tag{19}$$
$$b_{is} \geq \frac{1}{h}\sum_{l=1}^{h} x_{isl} \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S| \tag{20}$$

$$x_{isl} \in \{0,1\} \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S|,\ l=1,\dots,h \tag{21}$$

$$z_{isl}^1, z_{isl}^2, z_{isl}^3, z_{isl}^4 \in \{0,1\} \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S|,\ l=1,\dots,h \tag{22}$$

$$b_{is} \in \{0,1\} \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S| \tag{23}$$

$$y_{isl} \geq 0\ int \ , \forall\, i=1,\dots,n,\ s=1,\dots,|S|,\ l=1,\dots,h \tag{24}$$

The objective function (1) minimizes the handling costs (first component of the sum) and the products decentralization, mathematically expressed by the difference between the second and the third component. This for allocating the classes in such a way that, if possible, all the demand of a product type should be assigned to the same slot. Otherwise, according to the decentralization policy, the model allocates products in adjacent slots in the same block; in the same aisle; in opposite slots; in posterior slots and finally in generic slots, thanks to the fact that $p_1 > p_2 > p_3 > p_4$. The penalty p_5 is the highest value in order to avoid that the demand of a product is distributed in different locations. The constraints (2) impose the demand satisfaction for each class; the conditions (3)-(4); (6)-(7); (9)-(10) and (12)-(13) link the x variables to the relative z ones. The constraints (5)-(8)-(11) and (14) set to 0 the related z variables associated to the last level. The constraints (15)-(16) guarantee the compatibility conditions in slots of the same block and in the adjacent ones in the same aisle. The constraints (17) are the capacity constraints for each slot and the constraints (18) link the y variables to x ones. Finally, the constraints (19)-(20) link the x variables to the b ones. The last groups ((21)-(24)) of constraints are the binary, non negative and integer conditions on the relative variables, respectively. The reader is referred to Guerriero et al (2012) for more details.

3 A CLUSTER-BASED HEURISTIC FOR ALLOCATING PRODUCTS IN MULTI-LEVELS WAREHOUSES

This Section describes the cluster-based heuristic approach used for solving large scale instances, on which *Cplex* (the optimizer used for solving the model (1)-(24)) throws the *Out of Memory Exception*. The approach can be divided in 5 relevant steps: (step 1) to group the product classes into *clusters*; (step 2) to build an auxiliary graph *G*; (step 3) to find the connected components of *G*; (step 4) for each connected component of *G*, to build a relative tree throughout a depth visit and finally, (step 5) to allocate the classes by using the information derived by the trees.

In particular, the approach starts by grouping the products classes into *compatible clusters*. A *compatible cluster* only contains products compatible to each other. Of course, one class could theoretically belong to more than one cluster according to the compatibility matrix. For each couple of clusters, the *intersection* is computed.

Definition 2

Two clusters C1 and C2 have a not empty intersection if and only if at least two products classes $P1 \in C1$ *and* $P2 \in C2$ *are compatible to each other, that is* $C_{P1P2} = 1$.

By using the information on the intersection sets among the clusters, an auxiliary graph can be defined as a undirected graph *G=<V,E>*, where *V* is the set of vertices and *E* is the set of edges. In particular, for each cluster, an associated vertex *v* exists in *G* and for each couple of clusters/vertices (*u,v*), an associated edge exists in *G* if and only if their intersection is not empty (*Definition 2*). The third step provides the individuation of the connected components of the graph *G*.

Definition 3

Given a graph G, a connected component is a maximal sub-graph H.

Definition 4

A connected sub-graph H of G is maximal if and only if it is not possible to add in H other vertices and/or edges in such way that the new sub-graph H1 is still connected.

By using the *Definitions 3* and *4*, all the connected components are individuated.
The fourth step of the procedure builds a tree for each connected component, by executing a depth-first search. Each tree provides a possible sequence to be followed for allocating the classes into the warehouse taking into consideration the compatibility constraints. In fact, it is worth noting that, with respect to the problem in exam, each couple of connected components represents a *discontinuity* in the classes allocation that, anyway, cannot be avoided. The tree is always determined by a decreasing sort of the demands of its nodes. In particular, because each node of the tree is a cluster, its relative demand is the aggregated demand of all the classes that belong to it.

In order to improve the quality of the final solutions detected by the approach, we also perform, at the end of the procedure, a local search. In particular, we define two different moves: *switch move* and *remove move*. The first one, for each possible couple of classes *P1* in the current position *POS1* and *P2* in the current position *POS2*, tries to assign *P1* in *POS2* and *P2* in *POS1* and applies only the changes improving the quality of the current solution (reducing the cost). The latter, for each class *P1* in the current position *POS1*, tries to find a free slot and to allocate it to this new position. Again, it applies only the changes improving the quality of the current solution.

3.1 A toy example

In order to better illustrate the cluster-based heuristic for the problem in exam, this Section provides a simple example, focusing the attention on the most critical steps (i.e., (step 3) and (step 4)). Let's assume to have already grouped the products classes into different compatible clusters and to have individuated the relative connected components as depicted in **Figure 1**: *H=<V',E'>* and *H1=<V'',E''>* where *V'={C1,C2,C3,C4}* and *E'={(C1,C2), (C2,C3), (C1,C3), (C2,C4)}*, whereas *V''={C5,C6}* and *E''={(C5,C6)}*.

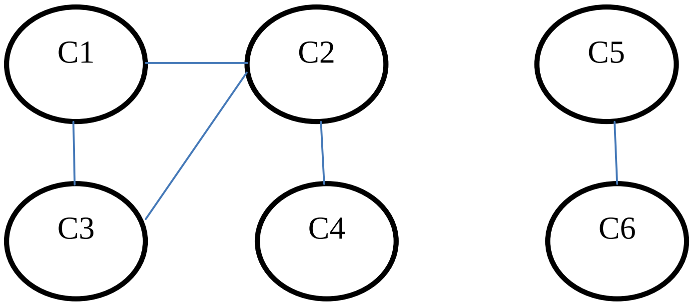


Figure 1. Connected components of the toy example

Focusing the attention on the first connected component *H*, let's assume that the total demand associated to *C1* is 10; to *C2* is 8; to *C3* is 6 and, finally to *C4* is 4. Thus, the depth visit, starting by the root *C1* (the highest demand node), provides the tree depicted in Fehler! Verweisquelle konnte nicht gefunden werden.**.** The second node to be added by the procedure is, among the remaining (*C2*, *C3* and *C4*), the one which has the highest demand. Thus, the procedure selects *C2*. After that, *C3* is added to the tree and at the same level *C4*.

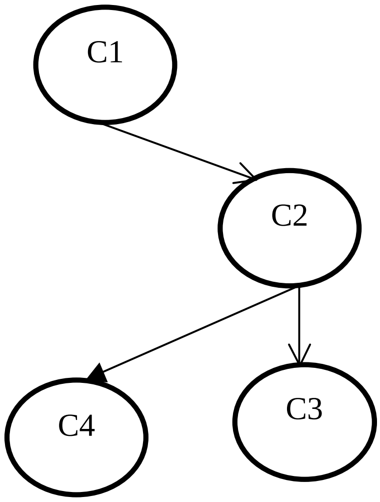


Figure 2. The relative tree of the connected component H in the toy example

The result depicted above gives a sequence for allocating the products belonging to the clusters of *H*. As it is possible to observe, one discontinuity exists in this specific case that cannot be avoided. Finally, considering the connected component *H1*, the resulting tree is the one depicted in **Figure 3**, assuming that the demand of *C5* is 10 and the one of *C6* is 6.

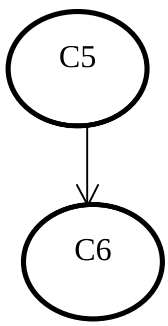


Figure 3. The relative tree of the connected component H1 in the toy example

4 COMPUTATIONAL RESULTS

This Section provides some computational results running both the cluster-based heuristic and the model. Both the model and the heuristic have been implemented using *Java* and, in particular, for the linear programming formulation, *Cplex* has been used as Optimizer. The test instances have been generated by using the procedure described in Guerriero et al (2012). In particular, four different testing sets have been individuated and considered for the experiments. The first one is generated by varying the slot capacity in the range [11,20,30,50,80,100,150,200] and fixing the number of products to 20. The second set is individuated by varying the number of layers in the range [4,5,6,7,8,9,10]; the third and the fourth set by varying the number of horizontal slots (in the range [6, 8, 10, 12, 14, 16, 18]) and the number of vertical slots (in the range [4,5,6,7,8,9,10]), respectively. For all the three above cases, the number of classes is firstly set to 50 and then to 80. For all the cases, two different scenarios have been taken into consideration: the former with few incompatibilities (for short *M scenario*) and the latter with a lot of incompatibilities (for short *C scenario*).

The numerical results have been obtained by comparing the solutions of the model with those determined by the proposed heuristic. These comparisons (in %) have been performed with respect to two criteria: the solution quality (in terms of cost) and the computational times. In particular, the final gaps in cost and in time have been computed referring to the solutions obtained by the local search. For example, the final gap between costs Gap^{final}_{cost} has been computed as in the following reported:

$$Gap^{final}{}_{cost} = \frac{\text{cost}_{\text{local search}} - \text{cost}_{\text{model}}}{\text{cost}_{\text{model}}} * 100 \quad (25)$$

where $\text{cost}_{\text{local search}}$ and $\text{cost}_{\text{model}}$ represent the cost of the heuristic approach after the local search and the cost of the model, respectively. In a similar way:

$$Gap^{initial}{}_{cost} = \frac{\text{cost}_{\text{cluster}} - \text{cost}_{\text{model}}}{\text{cost}_{\text{model}}} * 100 \quad (26)$$

where $\text{cost}_{\text{cluster}}$, instead, represents the cost detected by the heuristic procedure without the local search. This is done in order to show the improvements guaranteed by performing a local search on the solution. Same considerations of (25) and (26) can be applied for the gaps obtained between solution times. Moreover, the instances on which the model is able to provide only a feasible solution are remarked in bold in the relative tables, presented in the following. As shown by the following tables, by increasing the complexity of the instance (in terms of both the number of products to be allocated and the warehouse layout (i.e., number of storage locations)), the model is not able to detect optimal solutions (but only feasible ones). In these cases, the heuristic always outperforms the model providing better solutions (in terms of cost and computational time). However, in all the test cases, the heuristic outperforms the model with refer to the computational times (with a significant impact). This last aspect shows that the cluster-based heuristic, on the contrary of the model, is also able to detect good quality solutions for real scenarios in reasonable amounts of time. Moreover, the results underline a significant improvement of the local search on the final solution detected by the heuristic approach. For example, the average percentage value reported in

TABLE 3 highlights that applying the local search at the end of the heuristic guarantees a significant improvement of the final solution, considering that the initial gap in cost is about 9% and the final one downs to about 2%. These significant improvements are more visible increasing the complexity of the instances. Finally, in **Table 6**, two cases are remarked in grey and underline another important advantage of the proposed heuristic approach. In particular, on these instances, *Cplex* is not able to gather any information about the active model (both in terms of optimality and feasibility). Instead, the heuristic method detects final solutions in a reasonable amount of time.

Table 1. Varying cap (I)

	cost		**time**	
dataset	$Gap_{\cos t}^{initial}$	$Gap_{\cos t}^{final}$	$Gap_{time}^{initial}$	Gap_{time}^{final}
C_N20_CAP11	**-7.37**	**-7.82**	**-100.00**	**-100.00**
C_N20_CAP20	**0.51**	**-0.11**	**-100.00**	**-100.00**
C_N20_CAP30	**0.68**	**-0.12**	**-100.00**	**-100.00**
C_N20_CAP50	1.25	0.14	-100.00	-100.00
C_N20_CAP80	2.27	0.60	-100.00	-100.00
C_N20_CAP100	5.95	1.15	-100.00	-100.00
C_N20_CAP150	5.65	0.71	-100.00	-100.00
C_N20_CAP200	6.24	0.47	-100.00	-100.00
AVERAGE	**1.90**	**-0.62**	**-100.00**	**-100.00**

Table 2. Varying cap (II)

	cost		**time**	
dataset	$Gap_{\cos t}^{initial}$	$Gap_{\cos t}^{final}$	$Gap_{time}^{initial}$	Gap_{time}^{final}
MC_N20_CAP11	**-0.96**	**-2.67**	**-100.00**	**-100.00**
MC_N20_CAP20	**0.51**	**-0.07**	**-100.00**	**-100.00**
MC_N20_CAP30	**0.71**	**-0.12**	**-100.00**	**-100.00**
MC_N20_CAP50	1.47	0.16	-100.00	-100.00
MC_N20_CAP80	2.35	0.15	-100.00	-100.00
MC_N20_CAP100	7.83	1.61	-100.00	-100.00
MC_N20_CAP150	7.04	0.62	-100.00	-100.00
MC_N20_CAP200	9.15	0.92	-100.00	-100.00
AVERAGE	**3.51**	**0.07**	**-100.00**	**-100.00**

Table 3. Varying H (I)

dataset	cost		time	
	$Gap_{\cos t}^{initial}$	$Gap_{\cos t}^{final}$	$Gap_{time}^{initial}$	Gap_{time}^{final}
C_N50_H4	7.27	2.40	-100.00	-100.00
C_N50_H5	8.81	1.42	-100.00	-100.00
C_N50_H6	8.36	1.00	-100.00	-100.00
C_N50_H7	8.47	1.11	-100.00	-100.00
C_N50_H8	8.21	0.86	-100.00	-100.00
C_N50_H9	8.36	1.01	-100.00	-100.00
C_N50_H10	**6.44**	**-0.79**	**-100.00**	**-100.00**
MC_N50_H4	7.57	3.11	-100.00	-100.00
MC_N50_H5	10.04	2.03	-100.00	-100.00
MC_N50_H6	10.25	2.22	-100.00	-100.00
MC_N50_H7	10.08	2.07	-100.00	-100.00
MC_N50_H8	8.04	0.18	-100.00	-100.00
MC_N50_H9	9.56	1.59	-100.00	-100.00
MC_N50_H10	**7.77**	**-0.07**	**-100.00**	**-100.00**
AVERAGE	**9.12**	**1.80**	**-100.00**	**-100.00**

Table 4. Varying H (II)

dataset	cost		time	
	$Gap_{\cos t}^{initial}$	$Gap_{\cos t}^{final}$	$Gap_{time}^{initial}$	Gap_{time}^{final}
C_N80_H4	5.61	1.76	-100.00	-100.00
C_N80_H5	**7.59**	**-0.62**	**-100.00**	**-100.00**
C_N80_H6	**5.73**	**-1.88**	**-100.00**	**-100.00**
C_N80_H7	**5.33**	**-2.25**	**-100.00**	**-100.00**
C_N80_H8	**-19.18**	**-25.00**	**-100.00**	**-99.00**
C_N80_H9	**6.30**	**-1.35**	**-100.00**	**-99.00**
C_N80_H10	7.82	0.05	-100.00	-99.00
MC_N80_H4	6.86	1.56	-100.00	-100.00
MC_N80_H5	10.58	2.49	-100.00	-100.00
MC_N80_H6	**7.30**	**-0.55**	**-100.00**	**-100.00**
MC_N80_H7	**7.26**	**-0.58**	**-100.00**	**-99.00**
MC_N80_H8	10.09	2.03	-100.00	-99.00
MC_N80_H9	**7.50**	**-0.37**	**-100.00**	**-99.00**
MC_N80_H10	9.24	1.25	-100.00	-99.00
AVERAGE	**6.53**	**-1.04**	**-100.00**	**-99.00**

Table 5. Varying W (I)

	cost		tim.	
dataset	$Gap_{\cos t}^{initial}$	$Gap_{\cos t}^{final}$	$Gap_{time}^{initial}$	Gap_{time}^{final}
C_N50_W6	7.27	2.40	-100.00	-99.92
C_N50_W8	6.86	2.01	-100.00	-99.84
C_N50_W10	7.44	2.56	-100.00	-99.76
C_N50_W12	6.36	1.52	-100.00	-99.65
C_N50_W14	5.37	0.58	-100.00	-99.52
C_N50_W16	6.66	1.81	-100.00	-99.38
C_N50_W18	6.40	1.57	-100.00	-99.23
MC_N50_W6	7.57	3.11	-100.00	-99.91
MC_N50_W8	7.18	2.73	-100.00	-99.84
MC_N50_W10	7.23	2.78	-100.00	-99.75
MC_N50_W12	7.26	2.81	-100.00	-99.64
MC_N50_W14	7.66	3.20	-100.00	-99.53
MC_N50_W16	7.31	2.86	-100.00	-99.40
MC_N50_W18	7.29	2.84	-100.00	-99.22
AVERAGE	**7.34**	**2.42**	**-100.00**	**-99.61**

Table 6. Varying W (II)

	cost		time	
Dataset	$Gap_{\cos t}^{initial}$	$Gap_{\cos t}^{final}$	$Gap_{time}^{initial}$	Gap_{time}^{final}
C_N80_W6	5.61	1.76	-100.00	-99.84
C_N80_W8	**6.37**	**-1.16**	**-100.00**	**-99.65**
C_N80_W10	7.85	0.21	-100.00	-99.50
C_N80_W12	**3.71**	**-3.64**	**-100.00**	**-99.29**
C_N80_W14	**-29.36**	**-34.36**	**-100.00**	**-99.01**
AVERAGE	**-1.16**	**-7.44**	**-100.00**	**-99.46**
C_N80_W16				
C_N80_W18				
MC_N80_W6	6.86	1.56	-100.00	-99.83
MC_N80_W8	10.45	3.35	-100.00	-99.69
MC_N80_W10	**6.13**	**-0.70**	**-100.00**	**-99.49**
MC_N80_W12	9.95	2.88	-100.00	-99.23
MC_N80_W14	**-6.44**	**-12.46**	**-100.00**	**-98.98**
MC_N80_W16	**-20.28**	**-25.40**	**-100.00**	**-98.68**
MC_N80_W18	**-9.22**	**-15.06**	**-100.00**	**-98.33**
AVERAGE	**-0.36**	**-6.55**	**-100.00**	**-99.18**

Table 7. Varying M (I)

Dataset	cost		time	
	$Gap^{initial}_{cost}$	Gap^{final}_{cost}	$Gap^{initial}_{time}$	Gap^{final}_{time}
C_N50_M4	7.27	2.40	-100.00	-99.92
C_N50_M5	8.14	2.03	-100.00	-99.88
C_N50_M6	8.92	1.61	-100.00	-99.82
C_N50_M7	9.93	1.39	-100.00	-99.76
C_N50_M8	11.03	1.74	-100.00	-99.72
C_N50_M9	11.48	1.13	-100.00	-99.60
C_N50_M10	12.24	1.23	-100.00	-99.56
MC_N50_M4	7.57	3.11	-100.00	-99.92
MC_N50_M5	9.21	1.41	-100.00	-99.88
MC_N50_M6	10.11	2.36	-100.00	-99.82
MC_N50_M7	11.26	2.00	-100.00	-99.77
MC_N50_M8	12.42	1.89	-100.00	-99.72
MC_N50_M9	13.29	0.80	-100.00	-99.63
MC_N50_M10	13.97	1.15	-100.00	-99.56
AVERAGE	**10.81**	**1.79**	**-100.00**	**-99.76**

Table 8. Varying M (II)

Dataset	cost		time	
	$Gap^{initial}_{cost}$	Gap^{final}_{cost}	$Gap^{initial}_{time}$	Gap^{final}_{time}
C_N80_M4	5.61	1.76	-100.00	-99.84
C_N80_M5	6.95	0.46	-100.00	-99.75
C_N80_M6	**4.98**	**-1.83**	**-100.00**	**-99.63**
C_N80_M7	8.19	0.39	-100.00	-99.51
C_N80_M8	**8.34**	**-0.34**	**-100.00**	**-99.35**
C_N80_M9	**8.96**	**-1.00**	**-100.00**	**-99.18**
C_N80_M10	10.89	0.63	-100.00	-99.02
MC_N80_M4	6.86	1.56	-100.00	-99.84
MC_N80_M5	7.09	1.10	-100.00	-99.73
MC_N80_M6	**4.10**	**-2.21**	**-100.00**	**-99.63**
MC_N80_M7	8.60	1.09	-100.00	-99.52
MC_N80_M8	9.39	0.93	-100.00	-99.36
MC_N80_M9	11.04	1.31	-100.00	-99.18
MC_N80_M10	11.73	1.44	-100.00	-99.04
AVERAGE	**8.60**	**0.79**	**-100.00**	**-99.47**

5 CONCLUSION

In this paper, we have addressed the Products Allocation Problem in a multi-layers warehouse with compatibility constraints. In order to solve the problem in real settings, a heuristic approach has been defined. The proposed method, using the compatibility restrictions, groups the products in clusters and allocate the obtained compatible clusters to the storage locations.

In order to assess the performance of the proposed approach, in terms of solution quality and computational efforts, an extensive testing phase has been carried out, by considering a large set of problems. The computational results obtained are very encouraging and show that the proposed approach can be used to solve the Products Allocation Problem also in real scenarios.

REFERENCES

Guerriero, F., Musmanno, R., Pisacane, O. and Rende, F. (2012), A mathematical model for the Multi-Levels Product Allocation Problem in a warehouse with compatibility constraints, submitted to *Applied Mathematical Modelling*.

Heragu S.S. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research*, 43(2), 327-338, 2005.

Larson T. N., March H. and Kusiak A. (1997). A heuristic approach to warehouse layout with classe-based storage. *IIE Transactions*, 29, 337- 348, 1997.

Muppani, M. and Adil, G.K. (2005). Storage class formation considering space and handling cost for warehouse planning. *IX Annual International Conference of the Society of Operations Management*. Pondicherry University, Pondicherry, India.

Onout, S., Tuzkaya, U. R. and Dogac, B.(2008). A particle swarm optimization algorithm for the multiple-level warehouse layout design problem. *Computer and Industrial Engineering*, 54(4), 783-799, 2008.

Sanei, O., Nasiri, V., Marjani, M.R. and Moattar Husseini, S.M.(2011), A heuristic algorithm for the warehouse space assignment problem considering operational constraints: with application in a case study, *Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management*, Kuala Lumpur, Malaysia, 22- 24, January 2011.

SUPPLIER EVALUATION AND SELECTION DECISION MAKING METHODS

Nikolay Kolev

University of Ruse, Department of Management and Business Development, 8 Studentska St., 7017 Ruse, Bulgaria, nkolev@uni-ruse.bg

ABSTRACT

The paper deals with the procurement process as an integral part of logistics activities and more specifically with one of its core elements – supplier evaluation and selection. As business organisations are becoming increasingly dependent on their suppliers, and both direct and indirect consequences from erroneous decisions can be significant and affect the company's profitability, there is a need for systematic and transparent approach to handle partner evaluation and selection. Logistics procurement research in the past decades has proposed multiple methods and techniques for supplier evaluation but the complexity of the process requires further attention and investigation. Utilisation of appropriate methods can improve the effectiveness of decision making by means of proper structuring of tasks, formulation of relevant evaluation criteria and modelling of the selection process. On the basis of combination of existing procurement frameworks, the paper critically reviews established techniques, allocates them to the appropriate selection stages and proposes an appropriate combination of methods and techniques for the development of a complete model for supplier evaluation and selection.

Keywords: procurement process, supplier selection, decision making

1 INTRODUCTION

Supplier evaluation and selection problems have received significant attention in logistics research for the last several decades. Various decision making approaches have been proposed to address the problem but no consensus exists among researchers and practitioners with respect to the application of a unified approach. In addition, while it has been recognised that supplier selection is multi-objective in nature, little has been done to develop relevant methods capable of measuring suppliers' performance on multiple criteria.

The paper aims at filling this research gap by proposing suitable combination of existing quantitative techniques for supplier selection and evaluation and is organised as follows: Section 2 sets the basic elements of the procurement framework and discusses the various stages of supplier selection and the existing differences among purchasing situations. Section 3 provides a critical analysis of the methods and techniques applicable to the various stages of supplier selection, while Section 4 suggests and presents directions for future work together with summary and conclusions.

2 FRAMEWORK FOR SUPPLIER SELECTION

Procurement is a complex process extending over multiple stages, defined by major decision making situations. Aissaoui *et al* (2006) identify a logical framework, composed of six major procurement elements, coinciding with the main decision making processes, which can be applied to most of the purchasing situations: (1) a "make-or-buy" analysis, (2) supplier selection, (3) contract negotiation, (4) design collaboration, (5) procurement and (6) sourcing analysis.

A “make-or-buy” decision looks both at the economics of the part or service (i.e. cost-effectiveness analysis) and at its strategic importance to the enterprise. A company needs to decide on whether to produce a certain part or service internally or outsource it to an external supplier. This stage of the procurement process does not directly influence supplier evaluation and selection, but the outsourcing decision is a necessary prerequisite for future decisions. The supplier selection stage selects a group of suppliers for the delivery of products or services on a basis of predetermined criteria. This is one of the core decisions in procurement and many of the research papers focus on the selection of the best possible method for partner evaluation. Contract negotiation aims at designing appropriate contracts and clauses to provide certain security in the relationships with suppliers. This is also the stage to decide on the use of short- or long-term procurement contracts. Design collaboration stage is a similar process, which aims at the establishment of agreements between the purchasing organisation and the supplier to commonly develop parts and services that meet customer needs and specifications. Pure physical procurement decisions are related to securing timely and cost-efficient deliveries of products and services as well as to lot sizing and inventory management problems. In some cases, procurement decisions can be addressed as early as the supplier selection stage by using techniques capable of simultaneously selecting a supplier and allocation of total order quantities on the basis of evaluation results. The overall efficiency of the procurement process is addressed at the sourcing analysis stage. Aissaoui *et al* (2006) proposes that issues such as assortments, consolidation and supplier performance measurements would be considered at this final stage.

In general, only the supplier selection, procurement and sourcing analysis stages are within the sole competences of procurement departments and consequently require further investigation and analysis. All other phases of the procurement process require competencies and authority inherent either to other functional departments or top management.

Decision making at supplier selection level is not limited only to the final evaluation and selection of suppliers as proposed in the general procurement framework above. One appropriate structure, used in several contemporary research works is proposed by de Boer *et al* (2001), further amended and modified by Luo *et al* (2009) and Wu and Barnes (2011). It is composed of the following phases (1) supplier selection preparation, (2) criteria formulation, (3) pre-classification, (4) final selection, and (5) application feedback. Decision making at the preparation phase has the objective of defining the problem and identifying available alternatives as well as the desired results from the purchasing process which are not always obvious. Short product life cycles may necessitate the continuous search and selection of new and better suppliers while other purchasing situations, e.g. just-in-time, might require long-term partnerships. Criteria formulation stage requires that decision makers construct relevant sets of evaluation criteria on the basis of very specific organisational requirements in order to evaluate potential suppliers. Pre-classification is defined as a process for reducing the number of potential suppliers to a controllable level by classifying them into separate categories on the basis of their efficiency, thus making it a process of sorting rather than ranking. Research has indicated that pre-qualification is a necessary prerequisite for future successful supplier relations. Final supplier selection involves the application of detailed models to previously qualified partners for the choice of the most appropriate suppliers. This process might additionally involve order quantity allocation intended for cost optimisation along the supply chain, although this is usually a task for the procurement stage. Application feedback, as proposed by Luo *et al* (2009) is intended as an assessment process for the results from the previous stages by means of qualitative and quantitative methods. The application of principles for provision of feedback allows for continuous improvement and organisational learning.

The complexity of supplier selection is further amplified by the variety of existing purchasing situations. One appropriate classification is initially proposed by Kraljic (1983) and used with different modifications over the years. Purchasing situations are divided in three major groups - new purchasing tasks, modified purchasing tasks, and repetitive purchasing tasks for routine and strategic items.

New purchasing assignments are obviously the most complex situation characterised by high level of uncertainty – a new product or service needs to be procured and no prior experience or supplier information is available. Modified rebuy situations require the procurement of a new product from a known supplier or customised product from a new supplier. The level of uncertainty in these situations is moderate as either initial evaluation criteria are already identified and known (i.e. in case of existing products) or supplier performance data are easily available (in case of existing suppliers). Repetitive procurement has the lowest level of uncertainty as both product specifications and supplier performance information are existent and reliable. There is one notable difference in straight rebuy of strategic items where the supplier pool is extremely limited and often the evaluation situation is limited to appraisal and not an actual selection.

The main purpose of these classifications, illustrated in Table 1, is to provide purchasing organisations with a reasonable number of typical and manageable vendor selection scenarios with appropriate methods for carrying out and organising the evaluation and selection process.

Table 1. Supplier selection framework - adapted from Wu and Barnes (2011)

	New purchasing task	Modified purchasing task	Straight re-buy (routine products)	Straight re-buy (strategic products)
Supplier selection preparation	- Make-or-buy decision; - One-off decision; - Varying importance	- Decision on the number of suppliers to be used; - Moderate importance; - Repeating decision	- Replacing the current supplier; - Low importance; - Repeating decision;	- Decision on work changes with current supplier; - High importance; - Repeating decision
Criteria formulation	- No previously defined criteria; - Lack of historical data for suppliers performance	- Historical data available; - Previously used criteria available	- Historical data available; - Previously used criteria available.	- Historical data available but choice is limited; - Previously used criteria available
Pre-classification	- Large pool of suppliers; - Sorting rather than ranking	- Large pool of suppliers; - Sorting and ranking	- Large pool of suppliers; - Sorting rather than ranking	- Very small supplier pool; - Sorting rather than ranking
Final selection	- Smaller pool of suppliers; - Ranking rather than sorting; - Multiple criteria; - No historical records; - Model used once	- Small pool of suppliers; - Ranking rather than sorting; - Fewer criteria; - Lot sizing decisions; - Repeating use of model	- Small pool of suppliers; - Ranking rather than sorting; - Fewer criteria; - All quantities are allocated to a single supplier	- Very small supplier pool; - Evaluation rather than selection; - Sole souring
Application feedback	- Change of customer demands; - Decisions about model changes.	- Changes in supplier structure and quantities allocation	- Short or long term contracts; - Performance evaluation	- Improvements in the work with the current supplier

The characteristics of the above framework determine the differences in its application in the various possible scenarios. One possible example outlines the differences between new purchasing tasks and re-buy situations. In new assignments it is highly unlikely that a company has been in previous contacts with potential suppliers. This practically means that alternatives are evaluated on the basis of submitted offers and features of products and services. In re-buy scenarios, the relationships between suppliers and procurement organisations are closer, and selection decisions can also include different supplier characteristics such as manufacturing processes, employers, management structure, etc., rather than simple product/service specifications.

The general procurement framework, together with the supplier selection stages and types of purchasing tasks, described above define the main tasks that need to be solved in the process of supplier evaluation and selection. They require that decisions are taken separately at each level and a single method or technique might not be appropriate for every stage. Appropriate combination of methods has to be established at the levels of problem definition, criteria formulation, pre-qualification, final selection and application feedback in conjunction with the complexity of various purchasing situations in order to efficiently encompass the various purchasing decisions.

3 METHODS AND TECHNIQUES USED FOR SUPPLIER EVALUATION AND SELECTION

A large number of general multi-criteria decision making approaches have been adapted and used for supplier evaluation and selection, ranging from purely qualitative methods to complex operations research techniques. However, these techniques need to be analysed and allocated to each of the supplier selection framework stages in order to reach a conclusion about the appropriate combination of methods to be used in the vendor evaluation process. This analysis might not be applicable to each and every economic sector dealing with supplier selection, but in general it may serve as a guideline for procurement departments in their efforts to choose the most suitable partner for their operations.

3.1 Supplier selection preparation

Qualitative methods are predominantly used in the preparatory phase of supplier evaluation and selection. They aim to define the precise needs and requirements towards a potential supplier. As stated in Aissaoui *et al* (2006), this might not be an easy decision. Due to the shortened life cycles, continuous search for new suppliers is a priority for some companies, aiming to diversify their product range. In other purchasing environments, the establishment of long-term relationships with the supplier might be first priority requirement. Despite the importance of this stage of the supplier selection framework, research is very limited despite the fact that different purchasing situations require different decisions. No specific techniques can be identified and operations research methods are not applicable.

3.2 Criteria formulation

As a contrast to supplier selection preparation, the criteria formulation stage is well studied and research initiatives date back to the 1960s. The classic research works of Dickson (1966) and Weber *et al* (1991) are based on empirical investigation among purchasing managers and identify a number of criteria such as quality, delivery, performance history, warranties, production facilities and capacity, etc., each described by their relative importance. Contemporary procurement research also indicates that the categories of cost, quality, delivery, flexibility and customer service, each corresponding directly to manufacturing performance, continue to dominate at this stage of supplier selection.

The empirical qualitative methods are complemented by both theoretical and experimental methods, described in procurement research, but the selection of best criteria for a certain type of situation is often neglected. Expert’s judgement is widely used as opposed to more précised and objective techniques. As noted in Wu and Barnes (2011), there are certain models for generalisation of criteria, followed by specific modifications dependant on the type of industry. Huang and Keskar (2006) present a quantitative integration mechanism in terms of a set of comprehensive and configurable metrics arranged hierarchically that takes into account product type, supplier type, and OEM/supplier integration level. On the basis of a firm's business strategy, the management configures an appropriate set of metrics used to measure supplier performance. Dempster-Shafer theory (combination of evidence from different sources) is also applied to criteria formulation (Wu and Barnes, 2010). This work offers a rigorous and method with practical example for formulating criteria to use in partner selection decision-making in agile supply chains. The definitive drawback of quantitative approaches in criteria formulation lies in their difficult practical implementation in purchasing departments.

Several important conclusions can be made for the techniques used at this level of procurement decision making. Supplier selection criteria, used currently by purchasing organisations, are to a great extent identical to the ones identified in classical research works but their relative importance is changing among different purchasing situations and industries. A typical selection model would include both qualitative and quantitative criteria, often conflicting in nature, while their perceived importance would to a great extent be a product of subjective human judgement. In addition, new procurement tasks and repetitive tasks cannot make use of the same evaluation criteria. New assignments are based mostly on offers and/or product/service specifications while repetitive tasks can rely on additional data related to supplier characteristics and performance. The inclusion of broad range of criteria secures long-term relationships as factors such as design and technological capabilities can be introduced. However, the introduction of additional criteria makes selection decisions complex and problematic. Consequently, this stage of the procurement process requires clear differentiation of criteria to be used dependent on the type of industry and purchasing situation. Qualitative methods will probably remain dominant as there are numerous combinations of selection criteria in every purchasing situation and quantitative methods are too complex to be applied in the work of procurement organisations.

3.3 Pre-qualification of suppliers

The purpose of this stage is to rule out inefficient candidates, thus reducing the supplier pool to a manageable number of potential partners. Pre-qualification in almost all procurement situations represents a process of sorting rather than ranking. One notable exception is the purchasing of strategic products or services as the supplier pool is extremely limited and pre-qualification might not be necessary.

In vast majority of the cases, pre-qualification of suppliers relies on operations research and statistical methods, such as data envelopment analysis, cluster analysis and artificial intelligence models, represented by case based reasoning (Ho *et al*, 2010). These methods can also be used at the final selection stage, but their sorting nature makes them more suitable for pre-qualification.

Data envelopment analysis (DEA) is a non-parametric mathematical programming technique built around the idea of frontier analysis. It assesses the relative efficiency of decision making unites (in our case - suppliers) by calculating a ratio of weighted sum of outputs and inputs. As the model computes the most favourable set of weights for each supplier without making other suppliers' efficiency greater than one, DEA is able to overcome the subjective judgements for relative importance of criteria. Another major advantage of the technique is that DEA does not require *a priori* knowledge of the functional dependency between the inputs and outputs. In addition, DEA modifications such as introduction of assurance regions, super-efficiencies, variable or constant return of scale allow for significant flexibility in the process of pre-qualification. Assurance regions of multipliers (dual variables) (Thompson *et al*, 1990) allow for the incorporation of human judgement in the model, making it able to capture specific requirements of the purchasing organisation. Super efficiency iterations can further differentiate among weekly and strongly efficient suppliers, thus producing two distinctive groups of efficient and inefficient suppliers. As a non-parametric method, DEA is able to include both qualitative and quantitative evaluation criteria. Still, qualitative criteria depend on human judgement for their inclusion in the model. The most likely option for the inclusion of qualitative performance attributes is their evaluation via a Likert scale, later transformed into a convenient form for further calculations. One of the few disadvantages of DEA is the difficult differentiation between inputs and outputs. Multiple research works define the same performance attributes both as inputs and outputs. Since the functional dependency between variables is not required, in some cases the results from DEA may be infeasible. (Ho *et al*, 2010) estimate that approximately 18% of the research works on supplier selection between 2000 and 2008 have applied DEA as a primary operations research technique.

Cluster analysis (CA) is a statistical approach which uses an algorithm to classify suppliers into a number of clusters in such a way that the differences between items within a single cluster a minimal and differences between items from different clusters are significant. CA reduces the probability of rejecting a ‘good’ supplier too early in the process via subjective reduction of the often large original set (Aissaoui *et al*, 2006). In CA, suppliers and decision criteria represent objects to be clustered and characteristics to examine to find similarities between objects respectively (Bottani and Rizzi, 2008). A major drawback of CA is its computational difficulty and the possibility to solve the problem with different clustering algorithms which can lead to a different number of clusters while, in fact there are no significant differences between them.

Case base reasoning (CBR) is a part of artificial intelligence methods. As a general rule, they are based on computer-aided systems, which can be “trained” either by experts or by means of processing historical data. Subsequently, any person from the purchasing organisation - even without expert knowledge - can consult the system in similar but new decision situation. Similar to all artificial intelligence models, CBA represents a database that provides a decision maker relevant information from similar, previous decision situations and cases. CBR systems are very new and very complex and their efficiency drops with the increase in the number of cases (Wu and Barnes, 2011). In the classification of (Ho *et al*, 2010), CBR methods are used in 8.97% of the supplier selection cases.

Apart from the three major quantitative methods used in pre-qualification, qualitative methods are also applied at this stage of supplier selection. These techniques rely entirely on buyers’ experience and their subjective judgement, aiming to qualify potential partners in three major groups – “good”, “neutral” and “unsatisfactory”.

3.4 Final supplier selection

Most of the existing research concentrates on the final supplier selection stage. Models can be distinguished according to whether they are for single or multiple products’ procurement situations. Operations research methods dominate this stage of partner selection. Linear weighting models, mathematical programming models, analytical hierarchy/network process models and fuzzy set models are among the most often used techniques (Wu and Barnes, 2011).

Linear weighting models provide weights to criteria where the biggest weight indicated higher importance. Ratings are then multiplied by their weights and summed to obtain a single evaluation score. A lot of modifications have been introduced, including exclusion of compensatory models (where a high rating on one criterion can compensate a low rating on another) and introducing techniques for inclusion of uncertainty and imprecision. Simplicity of implementation is the main advantage of linear weighting models but despite the numerous adaptations they are unable to completely overcome subjective assignment of both weights and suppliers' scores. Any attempt to include objective assigning of weights and scores makes models too complicated for practical use.

Mathematical programming (MP) techniques applied to the problem of supplier selection include goal programming, multi-objective programming and integer programming. They aim to formulate the decision problem as a mathematical objective function which needs to be maximised or minimised. The main advantage of mathematical programming models is their ability to handle multiple products simultaneously, thus implementing lot-sizing and quantity allocation within the supplier selection phase. In addition, they can be easily combined with various techniques used at previous stages, using their results as inputs. Mathematical programming models are more objective but as opposed to DEA, they require explicit knowledge of the objective function and the dependencies between inputs and outputs. In addition it is very uncommon for MP techniques to include other than quantitative criteria in supplier selection problems.

Analytic hierarchy process (AHP) techniques provide a hierarchical structure to model the supplier selection phase. The hierarchy shows the relationships between goals, criteria and alternatives. These models allow the incorporation of uncertainty, experience and even intuition in a logical structure in order to calculate criteria weights as opposed to directly assigning them. However, AHP models are compensatory in their nature as the alternatives (suppliers) receiving low scores on a certain criterion can compensate with high score on another. As a difference to mathematical programming models, AHP can easily accommodate qualitative data, but assume uni-directional hierarchical factor structure, which is not always realistic. A further development of this technique, the analytic network process (ANP) allows for the inclusion of additional complexity and interrelationships between hierarchical levels and criteria. Additionally, ANP overcomes the problem of rank reversal, which is a common characteristic of AHP. As noted in (Sarkis *et al*, 2007), the final solutions of ANP may not be clearly defined without the introduction of secondary criteria.

Fuzzy set theory (FST) models can also be applied to the supplier selection problem since they allow simultaneous treatment of precise and imprecise variables (Wu and Barnes, 2011). FST can also model human judgement and incorporate multi-criteria information thus making it suitable for situations characterised with uncertainty. FST models can be successfully combined with linear weighting models to overcome inconsistencies in human judgement. However, the fuzzy set theory is complex and difficult for final users of the model.

4 CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK.

The various methods that can be applied to the decision making process for supplier evaluation and selection are by themselves a proof that no single techniques is applicable for all purchasing situations. An appropriate model needs to have some distinctive characteristics: (1) successful integration of qualitative and quantitative data; (2) evasion of subjective judgment while simultaneously providing an option for accommodating experts' knowledge; (3) ability to separate efficient and inefficient suppliers; (4) clear presentation and explicability to end-users (e.g. procurement departments). Thus, only the combination of several methods at different procurement stages can achieve these results.

One possible option to address the requirements is a combined DEA-ANP model for supplier selection. Such model will be able to include all major phases of decision making and can be easily adapted to existing software solutions for procurement management. Future research will be directed at the development of that model and gathering of empirical evidence for its feasibility.

REFERENCES

Aissaoui, N., Haouari, M. and Hassini, E. (2007), Supplier selection and order lot sizing modeling: A review, *Computers and Operations Research*, **34**, 3516-3540.

Bottani, E. and Rizzi, A. (2008) An adapted multi-criteria approach to suppliers and products selection – An application oriented to lead time reduction, *International Journal of Production Economics*, **111**, 763-781

De Boer, L., Labro, E. and Morlacchi, P. (2001), A review of methods supporting supplier selection, *European Journal of Purchasing and Supply Management*, **7**, 75-89.

Dickson, G. W. (1966), An analysis of vendor selection systems and decisions, *Journal of Purchasing*, **2**, 5-17.

Ho, W., Xu, X. and Dey, P. K. (2010), Multi-criteria decision making approaches for supplier evaluation and selection: A literature review. *European Journal of Operational Research*, **202**, 16-24.

Huang, S. H. and Harshal, K. (2007), Comprehensive and configurable metrics for supplier selection, *International Journal of Production Economics*, **105**, 510-523.

Kraljic, P. (1983), Purchasing must become supply chain management, *Harvard Business Review*, **61**, 109-117.

Luo, X., Wu, C., Rosenberg, D. and Barnes, D. (2009), Supplier selection in agile supply chains: An information-processing model and an illustration, *Journal of Purchasing and Supply Management*, **15**, 249-262.

Sarkis, J., Talluri, S., Gunasekaran, A. (2007), A strategic model for agile virtual enterprise partner selection, *International Journal of Operations & Production Management*, **27**, 1213-1234.

Thompson, R. G., Langemeier, L. N., Lee, C. T., Lee, E. and Thrall, R. M. (1990), The role of multiplier bounds in efficiency analysis with application to Kansas farming, *Journal of Econometrics*, **46**, 93-108.

Weber, C. A., Current, J. R. and Benton, W. C. (1991), Vendor selection criteria and methods. *European Journal of Operational Research*, **50**, 2-18.

Wu, C. and Barnes, D. (2010), Formulating partner selection criteria for agile supply chains: A Dempster–Shafer belief acceptability optimisation approach, *International Journal of Production Economics*, **125**, 284-293.

Wu, C. and Barnes, D. (2011), A literature review of decision-making models and approaches for partner selection in agile supply chains, *Journal of Purchasing and Supply Management*, **17**, 256-274.

ALGORITHM FOR FACTOR DETERMINATION OF INNOVATION ACTIVENESS OF MACHINEBUILDING SMEs (THE CASE OF BULGARIA, EXAMPLE OF RUSE REGION)

Svilen Kunev[1], Diana Antonova Antonova[1] and Svilena Svilenova Ruskova[1]

[1]*University of Ruse, Dept. of Management and Business Development, Bulgaria, 7017 Ruse, 8 Studentska str., snkunev@uni-ruse.bg, dantonova@uni-ruse.bg, svilena_ruskova@abv.bg*

ABSTRACT

The paper presents an algorithm for investigating the impact of factors influencing the innovation activeness of SMEs from machine-building sector in Ruse region. The approbation includes sequential defining the profiles of the enterprises under study, reflecting the interrelation between three main strategic variables, with the main purpose to describe the impact of factors influencing the innovation activity of the business entities while at the same time analyzing their effect on the strategic management and localization of the enterprises. The ***Initial profile*** organizes SMEs into four groups according to the directions of the innovation activity demonstrated: product innovators, organizational innovators, marketing innovators A and marketing innovators B. The ***Secondary profile*** of the innovation enterprises includes information about the innovation activeness of the enterprises under survey, giving an answer to the question which entities conduct innovation activity and how; which factors influence the innovation activeness and which ones affect the localization and the strategic directions of the business entities. The ***Integral profile*** of the innovative enterprises shows not only which factors have an influence on the enterprises with different innovation activeness, but also what their force of impact is. It distinguishes SMEs between technological innovators, product innovators, service innovators and process innovators.

Keywords: innovation activeness, SME

1 INTRODUCTION

The innovation activeness (IA) of enterprises is the factor determining the characteristics of the innovative product of business entities. This product, in turn, is the final stage of the innovation process, based on newly created and/or adapted existing knowledge and know-how, and it is materialized in the shape of new or significantly improved processes, products and services, whose final objective is the formation of competitive advantage (Andreeva, 2011; ARC Fund, 2010; Antonova, 2008). This necessitates the further development of approaches and methods for investigating of factors, affecting the innovation activity, in view of a more accurate study of the extent and strength of their influence.

The aim of the present paper is to present a methodological approach for investigating the impact of factors influencing the innovation activeness of SMEs from the sphere of machine-building in Ruse region. We present the logical sequence of the investigation, the results and conclusions. The approbation carried out includes defining the profiles of the enterprises under study in a three-stage sequence, reflecting the interrelation between three main strategic variables, with the main purpose to describe the impact of factors influencing the innovation activity of the business entities under study while at the same time analyzing their effect on the strategic management and localization of the enterprises.

2 ALGORITHM FOR FACTOR DETERMINATION OF INNOVATION ACTIVENESS

The innovation activeness is a comprehensive notion, based on quantity and quality indexes, defining the parameters of product innovations, generated as a result of innovation activity within the enterprise. In other words, the activeness is the final stage, the external manifestation of innovation activity (Kunev, 2010).

The measuring of innovation activeness is based on two main elements: (1) type of the innovations realized and (2) degree of novelty of the innovation solutions. On this basis, it is possible to characterize the subsequent activeness by quantity and quality with the help of absolute and relative values, various types of scales, statistical estimates and others. In brief, the information about the number and type of innovation solutions carried out by the enterprise shows the level of its innovation activeness.

A dividing line should be placed between innovation activeness and innovation intensity. According to some sources, the intensity can be viewed as a measurement of the degree of innovation activeness. It takes into consideration the ability of the enterprises to combine several types of innovations at the same time, in order to achieve more competitive advantages (ARC Fund, 2006). The intensity is also influenced by the degree of novelty, mostly of products that are launched on the market, i.e. unlike the innovative activeness, here the degree of novelty of the remaining innovations (process, organizational and marketing) is accounted to a lesser extent.

2.1 Approbation of methodological approach

The description of the algorithm proposed aims at defining the main steps and interdependence of the results obtained from the empirical study of the business entities from the sphere of machine-building in Ruse region. The main goal of the approach is to present a methodological tool kit for investigation of the innovation activeness of SMEs, its impact factors and the influence of some of them on localizing the business entities (Kunev, 2011). The following stages have been included:

Stage 1: Survey of localization factors.

At this stage the goals, objectives and subject of the investigation are defined; the statistical population is defined, as well as the way of forming a sample of respondents within this population; the method of conducting the survey, the tools, the surveying team, as well as the methods of analysis and presenting the results are defined. The aim of Stage 1 is to investigate the opinion of the business entities about the state of the localization factors in the region, as well as the degree to which these factors influence the innovativeness of the respondents surveyed.

STRATEGIC VARIABLE I is analyzed – Localizing and strategic planning of machine-building SMEs. *The investigation aims at determining which factors influence the localization, strategic planning and development of the business entities under survey*. The survey has been conducted among business entities from the sphere of machine-building in Ruse region. From a population of 200 acting entities a sample of 80 has been formed. The main contribution from the analysis at stage 1 is the formulation of ***Initial profile of innovative enterprises*** taking as ground the type of the business strategies applied. The enterprises are organized into four groups according to the directions of the innovation activity demonstrated: „product innovators", „oorganizational innovators", „marketing innovators A" and „marketing innovators B".

Stage 2: Survey of innovation activeness

STRATEGIC VARIABLE II is analyzed – Innovation activeness of machine-building SMEs. The research problem is to determine which factors have an impact on the innovation activeness of the different categories of innovative SMEs.

This stage complements the analysis and results from the previous one in the following aspects: while keeping the goal, objectives and population of the survey, an additional criterion to the entities under survey is introduced; the tool for conducting the survey is modified so that the thematic focus is defined more accurately. Based on the findings of Stage 2 a ***Secondary profile of the innovation enterprises*** is formulated, which includes information about the innovation activeness of the enterprises under survey, giving an answer to the question which entities conduct innovation activity and how; which factors influence the innovation activeness and which ones affect the localization and the strategic directions of the business entities.

Stage 3: Survey of factors influencing the innovation activeness

STRATEGIC VARIABLE III is analyzed – Intensity of the innovation activeness of machine-building SMEs. The research problem is to determine the force of impact of factors influencing the innovation activeness of the different categories of innovative SMEs.

During this stage the methodological analysis focuses on defining the extent of the force of impact that the influencing factors have. The outcome of the analysis at Stage 3 is an ***Integral profile of the innovative enterprises***, which shows not only which factors have an influence on the enterprises with different innovation activeness, but also what their force of impact is. We can distinguish between technological innovators, product innovators, service innovators and process innovators.

A general framework of the methodological approach is shown in Figure 1.

2.2 Results from the approbation

STRATEGIC VARIABLE I: Localizing and strategic planning of machine-building SMEs

PRODUCT INNOVATORS. Most important in their activity is availability of personnel with appropriate qualifications (69,2%); next is easy access to raw materials (61,5%). On the third place with equal results are presence of developed infrastructure and market potential of the region – 30,8% positive answers. Access to plots and buildings for setting up the company activities is important for 15,4% of the companies. With the lowest scores are the influence of local tax system (3,8%) and the opportunity for access to expert advice. In addition to evaluating factors' status, 69,2% declare that they prefer strategies for adopting a new product, mainly by investing in new and productive machinery.

ORGANIZATIONAL INNOVATORS. They also determine as most influential factor the qualified human resources (82,9%), access to raw materials (58,5%), infrastructure and market potential of the region – 39%, availability of plots and buildings – 17,1%, access to expert advice (9,8%) and the influence of local tax system (7,3%). Compared with the product innovators, a bigger part of this group of companies (78,1%) prefer to invest in highly productive equipment when exercise innovative activity.

MARKETING INNOVATORS "A"(increasing the market share). Most influential factor is the qualified human resources in the region (72,7%), access to raw materials (51,5%), market potential of the region (33,3%), well-developed infrastructure (24,2%). The availability of plots and buildings is important for 15,2% of companies; the status of the of local tax system has rating of 9,1%; access to expert advice has a result of 6,1%. Investing in technological equipment for performing the innovative projects is a strategic goal for 63,6% of the respondents.

MARKETING INNOVATORS “B”(entering new markets). Here equally important are the presence of qualified human resources in the region and the access to raw materials – each of those factors has 66,7% result. Next is market potential of the region (48,1%), followed by well-developed infrastructure (44,4%). The access to expert advice is a little bit more important than the other groups of innovative companies – 14,8%. The availability of plots and buildings is important for 11,1% of companies and the status of the of local tax system has rating of 7,4%. This profile has the lowest share of companies that could invest in new highly-productive equipment – 59,3%.

STRATEGIC VARIABLE II: Innovation activeness of machine-building SMEs.

Major part of PRODUCT INNOVATORS (72,7%) declare that they have experience in performing innovation project and will innovate in the future preferably in products (81,2%) and technologies (63,4%). If they decide to have partners for innovation project they would like to work with supplier, customer, consultant or international company – 45,5% for each partner. When starting innovation activity they are most influenced by the customer preferences (90,9%), the legislation, intellectual property rights, the presence of qualified human resources in the region, and access to supportive programs – 45,5% for each factor.

A lot of ORGANIZATIONAL INNOVATORS (81,8%) are experienced in developing innovative project in the past, 63,6% say that they would like to innovate in products in the future, while in technologies and processes will innovate equal parts of respondents – 45,5% for each type. A consultant and international company are the most preferred partners (45,5% for each). This is the difference with the previous group of innovative companies, because eventual partnerships with supplier and customer are put on the third and fourth places. The influential factors are similar, the strongest impact have the customer preferences (81,8%), followed by intellectual property rights protection, the presence of qualified human resources in the region and the legislative framework. In contrast to product innovators, organizational innovators do not pay big attention to access to supportive programs – they have result of 27,3%.

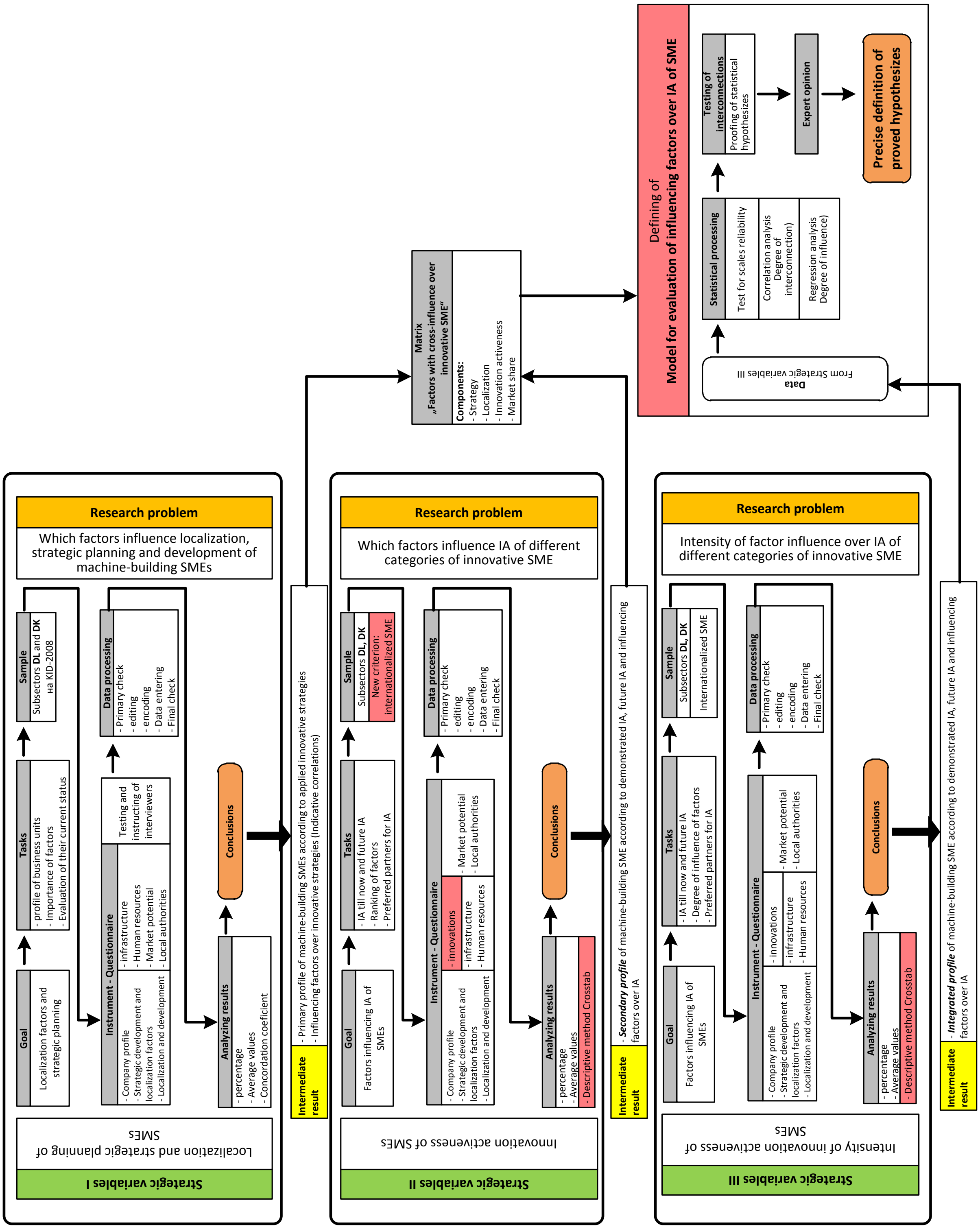


Figure 1. Framework of algorithm for factor determination of innovation activeness of machine-building SMEs

Three thirds (75,0%) of MARKETING INNOVATORS TYPE ”A” have demonstrated innovation activeness to the moment. In the future most of those companies will focus their efforts on designing new products (65%), but the other ways to innovate – in technologies, services and processes – are almost equally represented (40-45%), which shows that the business units from that profile strive for a balanced performance in the whole range of innovation opportunities. Preferences for partner in innovation activities do not differ very much from the other two profiles: higher scores have the suppliers (45%), international company (45%), client (40%) or consultant (40%). The innovative projects of this type of marketing innovators are affected mainly by customers' preferences (85%), the presence of qualified human resources in the region (40%), cooperation with a partner and the legislative framework (35% each). Relatively important factor is also the access to financial resources for funding innovation projects (35%).

Big amount (85,7%) of MARKETING INNOVATORS TYPE ”B” declare that they have previous experience in innovative projects. That is the highest score among all other innovative profiles. Here also the most liked options for future innovative projects are product innovations (71,4%), but the second place is for the innovations in services (57,1%). The partner with highest level of approvement are the consulting agencies (42,9%), followed by suppliers and international companies with 35,5% result. Clients a fourth in the selection with rate of 28,6%. Customer preferences and the presence of qualified human resources in the region exert big influence on that group of innovators – 85,6% and 57,1%. Next are access to supportive programs (42,9%), partner for evaluation (35,7%) and the possibilities for protection of intellectual property rights (28,6%).

STRATEGIC VARIABLE III: Intensity of the innovation activeness of machine-building SMEs

For TECHNOLOGICAL INNOVATORS most of the reasons for a starting a business have strong impact – half of the group answer that are heavily influenced by the practical application of skills and knowledge and protecting the personal and family prosperity; other 35,7% determine as strong the influence of increase in incomes as a factor. From the factors of strategic planning very strong impact have only the presence of appropriately qualified human resources and the status of the local tax system. With strong influence are the factors availability of plots and buildings (50%) and the market potential of the region (42,9%). From the group of innovation factors most of the respondents point that strong impact have the customers' interest for new products and services (64,3%) and the favorable national legislation (50%). For half of the companies the presence of a partner for innovation projects is with average influence, and another half of units perceive the type of impact of establishing clusters as little. From the group of localization factors five have strong impact – access to raw materials, access to financial services and resources, small number of competitors, presence of qualified human resources, and availability of plots and buildings. Factors with average impact are the proximity to railway stations and ports, the positive social climate and the well-developed transport infrastructure.

Most of PRODUCT INNOVATORS think that strong impact in starting the business activity has the practical application of skills (66,7%) and the generation of incomes (55,6%). Strategic planning is very strongly influenced by the presence of appropriately qualified human resources (55,6%) and the market potential of the region (44,4%); strong impact have access to raw materials (55,6%), and availability of plots and buildings (44,4%). The choice of location has strong impact (44,4%), together with the direct access to markets for production (33,3%). A lot of the innovation factors have very strong impact – such are the governmental support (66,7%), the educational system (44,4%). From the market factors very strong impact has the entering of new corporate clients in the region (44,4%), and the same amount of companies perceive the influence of improving the standard of living as strong. Among the localization factors very strong impact has the presence of qualified human resources (44,4%), and for 33,3% of respondents strong influence have the factors proximity to airports, railway stations and ports, the positive social climate, the well-developed transport infrastructure, presence of universities in the region and the low local taxes.

The SERVICE INNOVATORS are very strongly influenced (41,7%) by the practical application of skills and knowledge and protecting the personal and family prosperity when starting a business. Relatively strong is the impact of the need to generate of incomes (41,2%) and frustration from previous occupation (29,4%). When performing their strategic planning companies perceive the influence of the presence of appropriately qualified human resources as very strong (58,8%), and strong impact has the easy access to raw materials (41,2%) together with the market potential of the region (35,3%). As for the group of innovation factors the customers' interest for new products and services has very strong impact for 70,6% of the companies; the same is with the availability of qualified human resources, the active governmental programs for supporting innovations (47,1%). From the localization factors only the presence of qualified human resources has very strong impact (47,1%); strong is the influence of the positive social climate and the well-developed transport infrastructure (29,4%).

The PROCESS INNOVATORS also are very strongly influenced (41,7%) by the practical application of skills and knowledge and protecting the personal and family prosperity when starting a business. The same number of respondents agree that the presence of appropriately qualified human resources exerts very strong influence on their strategic planning; strong is the impact of easy access to raw materials (50%), access to plots for location (41,7%) and the market potential of the region (50%). The choice of location has average impact, while the direct access to markets for the production has strong impact on the decision for the initial location (41,7%). A lot of innovation factors have very strong impact. That is most visible with the customers' interest as a factor (66,7%). The difference in that group of innovators compared with the other profiles is that they evaluate as very strong the impact of presence of appropriate partner for innovation project (41,7%). Another special feature is that the presence of qualified human resources has undefined influence, which is between strong and very strong impact. From the market factors only the increase of incomes and the standard of living has strong influence. There are no location factors with very strong impact, only two of them have strong type of influencing forces - the proximity to airports, railway stations and ports (41,7%), and the availability of qualified human resources in the region (50%).

3 CONCLUSION

The main goal of the 3-stage study is to formulate analytical characteristics of machine-building SMEs in Ruse region (Bulgaria), focused on defining the innovation activeness and the factors, that have influence. For that reason on every step of the study a specific type of innovative profile is elaborated, combining several major parts of the business activity of the companies under survey.

The **Initial profile** of innovative enterprises takes as ground the type of the business strategies applied. The enterprises are organized into four groups according to the directions of the innovation activity demonstrated: „product innovators", „oorganizational innovators", „marketing innovators A" and „marketing innovators B". The results are:

- With product innovators there is a positive dependence among the factors "availability of raw materials", "access to land and buildings" and „developed infrastructure in the region".
- With organizational innovators the following factors are relatively more significant compared to outcomes from other factors: „availability of qualified human resources" "developed infrastructure", „access to plots and buildings" and „availability of raw materials".
- With marketing innovators from group A the only factors with positive relation to their change of innovation activity are: „access to land and buildings" and „level of the local taxation system". The remaining factors under survey have a negative mark of

interaction, with the correlation coefficients of "influence of developed infrastructure" and "access to expert aid" being more significant than others;

- The innovation activity of marketing innovators from group B correlates mostly with the changes in the state of infrastructure; the changes of the region's market potential; the access to raw materials; the possibility for using expert advice.

The **Secondary profile** of the innovation enterprises includes information about the innovation activity of the enterprises under survey, giving an answer to the question which entities conduct innovation activity and how; which factors influence the innovation activity and which ones affect the localization and the strategic directions of the business entities. The results from that stage show that:

- All four groups of enterprises have had experience in implementing innovation projects so far, with marketing innovators B having the highest value (85,7), followed by organizational innovators A (81,8%), marketing A (75%) and product innovator (72,7%);
- Product innovations are most preferred for future innovation processes, not surprisingly achieving the highest percentage of approval with product innovators (81,2%), followed by market innovators B (71,4%);
- The most preferred partners for joint innovations are supplier, client, international company or consultant. In any case the client's desire is a priority factor for all four categories of innovators.

The **Integral profile** of the innovative enterprises shows not only which factors have an influence on the enterprises with different innovation activeness, but also what their force of impact is. We can distinguish between technological innovators, product innovators, service innovators and process innovators. Results:

- When starting a business activity, the practical realization of skills and abilities, together with the guarantee of personal and family security, have a very strong influence on the biggest part of the four categories of SMEs – from 41,7% for the process and operational innovators, through 50% for the technological innovators, to 66,7% for the product innovators;
- The availability of work force with appropriate qualification has a very strong influence on the strategic planning of business entities;
- The interest of clients towards new goods/services, the favorable national legislature, the protection of intellectual property, the measures for encouraging innovations are factors with an extremely strong influence for a prevailing part of SMEs from different categories;
- Raising the living standard and income in the region has a medium to strong influence as a market environment factor;
- With the localization factors, strong to very strong influence characterizes the availability of qualified work force in the region – this is most significant for service innovators (47,1%) and product innovators (44,4%). Proximity to airports, railway stations and ports, as well as developed infrastructure displays medium to strong influence (28,6-41,7%).

REFERENCES

Andreeva, M. *et al* (2011), *Study of innovation alternatives for business survival and development*, Published under project FNI-2011-FBM-2, University of Ruse, Ruse, 2011. ISBN 978-954-8675-26-0.

Antonova, D. (2008), Technolgical innovations and post-industrial society, *Proceedings 6th International Scientific conference "Management and Engineering"*, **volume 1(104)**, Technical University, Sofia, June 2003, p. 336-341, ISSN 1310-3946.

ARC Fund (2010), *Innovations BG – 2010. The Bulgarian Innovation policy – opportunities for the next decade*. Applied research and communication foundation, Sofia, p. 17, ISSN 1313-1052.

ARC Fund (2006), *Innovations BG – the Bulgarian Innovation in European Union*. Applied research and communication foundation, Sofia, 2006, p. 17, ISBN 954-9456-08-0.

Kunev, S. (2011), Methodical approach for study of influential factors on innovation activeness of small and medium enterprises. *Proceedings 9th International Scientific conference "Management and Engineering"*, **volume 2(122)**, Technical University, Sofia, 19-22 June 2011, p. 685-692, ISSN 1313-7123.

Kunev, S. (2011), Influential factors on innovation activeness of small and medium enterprises – approbation of methodical approach of study,

Kunev, S. (2010), Main characteristics of innovation activeness in business orientated economic units, Proceedings 8th International Scientific conference "Management and Engineering", **volume 3(313)**, Technical University, Sofia, June 2010, p. 336-341, ISSN 1310-3946.

METAHEURISTIC APPROACH FOR FUZZY TRANSPORT PLANNING

Julio Brito Santana[1] and José A. Moreno Pérez[1]

[1]*Departamento de Estadística, I.O. t Comkputación, Instituto Universitario de Desarrollo Regional, Universidad de La Laguna. 38271 La Laguna. Spain, jbrito@ull.es, jamoreno@ull.es*

ABSTRACT

Transport route planning is one of the most important and frequent activities in supply chain management. The design of information systems for route planning in real contexts faces two relevant challenges: the complexity of the planning and the lack of complete and precise information. The Fuzzy Set Theory provides a suitable methodological approach for dealing with uncertainty which is a product of the imprecise nature of the information and decisions. Metaheuristics and Hyper-heuristics are optimization tools appropriated for complex problems while dealing with the flexibility of some of their components. The Soft Computing approach integrates specific fuzzy-based methodologies with the flexibility of the heuristic procedures for providing Intelligent Information systems for the development of transport route planning in uncertainty decision making contexts. Several fuzzy optimization models can be used to address the imprecision and/or flexibility in the vehicle routing problem formulations. These problems are then solved using hybrid metaheuristics combines their features that are capable to adapt to dynamic changing environment by using hyperheuristic based collaborative approaches.

Keywords: Transport Route Planning, Metaheuristics, Fuzzy Approach, VRP

1 INTRODUCTION

One of the missions and functions of commerce, logistics and transport is to bring together disperse or specific supply with an unequally distributed demand from a geographical perspective. A second mission is to organize the transfer of goods during a chain of transactions that reaches different destinations in an optimal way. Logistics operations include all planning and management activities that are carried out in the distribution channels which allow products to arrive under specific conditions of quantity, time, quality and location to satisfy market demands. From the operational point of view, routing problems constitute an important element of logistics systems. One of the most frequently studied routing problems is the Vehicle Routing Problem (VRP) (Cordeau *et al.*(2007)).

Technological developments are providing important advances in the availability of information systems that are able to support planning and management. However, support systems for logistics and transport planning must deal with the increase of the complexity of their development in a dynamic environment of rapid changes and high levels of uncertainty. Most of the transport parameters and decisions are characterized by undetermined, incomplete or unavailable information; and subjectivity, ambiguity or imprecision in descriptions and interpretations of the decision makers. In practical real-world problems decision makers use subjective knowledge or linguistic information when making decisions, measure parameters, objectives and constraints and even when modelling the problem (see Teodorovic (1999)).

Classic methods and models based on numerical analysis and exact or binary value logic are characterized by their precision and categorization. However this approach, included in hard computing, although exact, is not always the most practical way to solve route transport problems. On the other hand, techniques based on Intelligent Computing, include Soft

Computing (e.g. fuzzy logic, artificial neural networks, metaheuristics, etc.), are useful in route transport planning because they are flexible enough to deal with complex systems, offer acceptable approximate solutions and therefore add value. The advantages of employing Soft Computing in real-world problems are its capability to tolerate imprecision, uncertainty, and lack of information to achieve tractability and robustness in decision making with low costs (see Avineri (2005), Tiwari and Mehnen (2010), Verdegay *et al.* (2008)). Traditionally, the uncertainty found in the nature of the data and their settings has been handled by means of probability theory (Viertl and Hareter (2004)). When some elements of the problem are stochastic or random, researchers usually specify them as Stochastic Vehicle Routing Problems (SVRP), modelled either as a chance constrained program (CCP) or as a stochastic program with recourse (SPR) (see Gendreau *et al.* (1996), Stewart and Golden (1983), Tillman (2002)). In many practical cases uncertainty cannot be considered random phenomena and therefore probability theory cannot be applied successfully. Bellman and Zadeh (1970) described and named this type of imprecision, stochastic uncertainty, in contrast with the semantic meaning of imprecision which is appropriate to model judgments, preference and values and which cannot be used to estimate exact numeric values, since they are vague or fuzzy. Most VRP models with fuzzy components that have appeared in the literature assume vagueness in customer demand (Liu and Lai (2009), Lucic and Teodorovic (2003, 2007), Teodorovic and Kikuchi (1991)), and fuzzy times: service time and travel time parameters (Guo and Li (2006), Hong and Xu (2008), Hong and Xu (2008), Jia *et al.* (2008), Tang *et al.* (2007), Viertl and Hareter (2004)). Only a few references also consider uncertainty in time windows for service (Djadane *et al.* (2006), Kuo *et al.* (2004)). In addition, some authors argue that it is possible that randomness and fuzziness are present at the same time. In this situation, an integration of probability and possibility theories are used to deal with this type of hybrid uncertainty (Cheng and Gen (1995), Erbao and Mingyong (2009), Teodorovic and Pavkovic (1996), Zheng and Liu (2006)).

VRP are a well known combinatorial optimization problems included in the category of NP-hard problems. VRP can be solved by employing exact mathematical methods that guarantee finding an optimal solution if it exists. However, the required computational complexity leads to exponential processing time when the problem size is large. In this case, heuristic and metaheuristics methods are more suitable. The imprecision in some of the formulation components we can express it with fuzzy terms then we are faced with fuzzy optimization problem. Fuzzy optimization models with metaheuristics include advantages for modelling real problems with uncertainty in information.

2 ROUTE PLANNING IN UNCERTAIN CONTEXTS

Distribution and transport processes require decisions to be made practically on a daily basis, if not in a completely dynamic sense. These decisions concern specific routes that the transport units need to follow to distribute materials and products between different points in the supply chain. The establishment of routes that follow these units for the delivery or collection of goods is the central part of operational decision making in logistics management for many businesses.

Problem characteristics and decision making in general and specifically in transport planning are conditioned by resources, objectives, constraints and the nature of the information; these components influence model formulation and solution methods (Ganesh *et al.* (2007)). Transport planning problems require knowledge of the transport modes, the characteristics of the road network, the fleet of mobile media, customer or demand points and the distribution centres (warehouses, depots, platforms, transit centres, ports, airports and even other mobile media or platforms). These problems also need to determine the route(s) for each depot and the available fleet that satisfy the set of established constraints while trying to attain the proposed objectives.

Planning objectives are established by the decision maker, considering different criteria, such as:

- minimize fixed and variable costs,
- minimize fleet size,
- minimize total transport times and/or total distance travelled,
- minimize waiting times,
- maximize operating profits,
- maximize the load capacity of each vehicle,
- maximize the utility function and customer satisfaction.

Several different sources of uncertainty in transport are present in real world contexts. In general users and decision makers diligently establish measurements based on observations and perceptions which determine the problem parameters and in the same way affect the evaluation of objectives and obtained solutions.

The relative uncertainty found in measuring time and distance is one of the important phenomena related to transport planning. In many situations exact time and distance between two nodes is not known or expressed with precision. Examples are seen with the travel time between two nodes, which can be influenced by traffic congestion or road conditions and usually distance information in map or cartography available is not sufficiently precise or updated. It can also be difficult to specifically express customer service time, unload times, down times and waiting times. It is possible that customers want to establish their own service time limits, and these are normally expressed with preferences that are ambiguous, flexible, or imprecise.

Another source of uncertainty in transport activities is customer demand. A common dilemma facing customers is to indicate a specific demand with sufficient advance time, especially when stating the exact demand or expressing proper and understandable units with standardized measures. This situation creates an environment of uncertainty regarding demand for the decision makers.

In addition it is difficult to establish with precision vehicle loads and capacities in relation to the different types of loads that can be transported and the ways that they are measured. In this respect the standardization of packing and packaging is not sufficient, given the wide variety of product types and containers found in the vehicles.

On the other hand, there are problems where the decision maker needs to define a complex cost function which has information regarding transport mode, used vehicles and specified route parameters, many of which are imprecise or incomplete. Finally, we point out that there are also problems where all of the objectives and constraints are not necessarily strict or need to be exactly satisfied with the same precision. In other words it is possible to establish problems where the objectives are not strictly attainable and/or where constraints are not completely satisfied but with the same degree of precision.

We assume uncertainty is an important part of decision making, intrinsic for most real world routing planning problems that cannot be ignored. We use a fuzzy approach to deal with a type of imprecision associated with the vague and imprecise nature of linguistic terms that are used in the problem. In addition, this approach permits greater tolerance in the evaluation of objectives and constraints. In both cases the specification of probabilities and the simulation are not possible or are very expensive.

3 HEURISTIC OPTIMIZATION APPROACH

Solution methodologies for the Transport Routing Planning Problem variants include both exact and heuristic techniques. The inability of exact approaches to solve medium and large scale vehicle routing problems, as well as the difficulty in evaluating the objective function in real-life complex problems are two important reasons why heuristics and metaheuristics are mainly employed for the solution of these problems. Different methods have been developed, including Greedy Randomized Adaptive Search Procedure (GRASP) (Resende and Ribeiro (2003)) and Variable Neighborhood Search (VNS) (Hansen *et al.* (2010)) for most of these problems Bräysy y Gendreau (2005), Gendreau *et al.* (2008). In addition, Ant Colony Optimization (ACO) (Dorigo and Stützle (2003)) and Particle Swarm Optimization (PSO) (Clerc (2005)) strategies can be used to tune the parameters approach in a learning mechanism that converts the whole approach in a Hyper-heuristic method.

We propose a two-phase hybrid metaheuristic composed by a GRASP construction phase and a VNS improvement phase. The main GRASP parameters are those to manage the Restricted Candidate List (RCL) and the main VNS parameters are those controlling the set of the neighbourhoods. Let S and T be the corresponding sets of parameters of both phases.

The ACO and PSO based hyperheuristics are swarm inspired process where each individual (ants or particle) is represented by a setting (s,t) of the parameters for the GRASP/VNS that is applied to route planning. The individuals in the swarm use the performance of the corresponding solution procedure to cooperate; usually the total cost of the solution obtained $c(s,t)$. The ACO approach is a centralized cooperative scheme based on pheromone trails. The PSO approach is a decentralized approach where individual use social relationship to cooperate by an attraction mechanism.

For a preliminary analysis, we consider the following simplest setting for the low level heuristics that uses couple of single parameters (s,t). The GRASP phase incorporates to the partial solution a random customer from RCL that is obtaining by including the s nearest customers to end points of the partial routes; i.e. s is the RCL size. The VNS is a Variable Neighbourhood Descent (VND) that considers the k-chain neighbourhoods, for $k \leq t$; i.e. t is the number of neighbourhoods. A k-chain move consists of inserting a chain of k consecutive points of the solution in another position.

ACO uses the relative objective $(c_{max} - c(s,t))/(c_{max} - c_{min})$ to increase the pheromone trail of each pair (s,t) that are normalized as evaporation mechanism. The particles of PSO use their own best pair, the best neighbour pair and the global best pair as attractors. The roulette wheel rules selects the attraction move that consists in moving the two parameters towards the attractor values.

These approaches have been applied to solve instances of a real–life based optimization problem provided by distribution companies.

4 CONCLUSION

The Soft Computing approach integrates specific fuzzy-based methodologies with the flexibility of the heuristic procedures for providing Intelligent Information systems for the development of transport route planning in uncertainty decision making contexts. The Fuzzy Set Theory provides a suitable methodological approach for dealing with uncertainty which is a product of the imprecise nature of the information and decisions. Metaheuristics and Hyper-heuristics are optimization tools appropriated for complex problems while dealing with the flexibility of some of their components. GRASP/VNS hybrid metaheuristics are suitable for combining construction and improving phases that can be including in a learning approach to improve performance and adaptation by using ACO and PSO strategies for design hyperheuristic method for the hybrid.

ACKNOWLEDGEMENTS

This research has been supported by the projects TIN2008-06872-c04-01 (70% FEDER) from the Government of Spain.

REFERENCES

Avineri, E. (2005), Soft Computing Applications in Trafic and Transport Systems: A Review, *Advances in Soft Computing* **1**, 17-25.

Bellman, R.E. and Zadeh, L.A. (1970), Decision Making in a Fuzzy Environment, *Management Science* **(17B)4**, 141-164.

Bräysy, O. and Gendreau, M. (2005), Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Part II: Metaheuristics. *Transportation Science* **39(1)**,104-139.

Cheng, R. and Gen, M. (1995), Vehicle routing problem with fuzzy due-time using genetic algorithm, *Japanese Journal of Fuzzy Theory and Systems* **7(5)**, 1050-1061.

Clerc, M. (2006), *Particle Swarm Optimization*. ISTE (International Scientific and Technical Encyclopaedia).

Cordeau, J-F., Laporte, G., Savelsbergh M. and Vigo D. (2007), Vehicle Routing, *Handbook in OR & MS*, 14 (6), 367-427.

Djadane, M., Goncalves, G., Hsu, T., and Dupas, R., (2006), Dynamic Vehicle Routing Problems under Flexible Time Windows and Fuzzy Travel Times, *Proceedings of 2006 International Conference on Service Systems and Service Management* **2**, 1519-1524.

Dorigo, M. and Stützle, T. (2003), The ant colony optimization metaheuristic: algorithms, applications, and advances, in: F. Glover, G. Kochenberger (Eds.), *Handbook of Metaheuristics*, Kluwer.

Erbao, C. and Mingyong, L. (2009), A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands, *Journal of Computational and Applied Mathematics* **231**, 302-310.

Ganesh, K. Nallathambi, A.S. and Narendran, T.T. (2007), Variants, solution approaches and applications for Vehicle Routing Problems in supply chain: agile framework and comprehensive review, *International Journal of Agile Systems and Management*, **2(1),** 50-75.

Gendreau, M., Laporte, G. and Seguin, R. (1996), Stochastic vehicle routing. *European Journal of Operational Research* **88(1)**, 3–12.

Guo, J.Y. and Li, J. (2006) A Hybrid Genetic Algorithm to the Vehicle Routing Problem with Fuzzy Traveling Time, *Journal of Industrial Engineering Management* **19**, 13-17.

Hansen, P., Mladenović, N. and Moreno, J.A. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research* **175(1)**, 367-407.

Hong, L. and Xu, M. (2008a), A model of MDVRPTW with fuzzy travel time and time-dependent and its solution, *Proceeding of Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, **3**, 473-478.

Hong, L. and Xu, M. (2008b), Real Vehicle Routing and Dispatching with Dynamic Fuzzy Travel Times, *Proceeding of Second International Conference on Genetic and Evolutionary Computing*, 32-37, doi:10.1109/WGEC.2008.28.

Jia, J., Liu, N. and Wang. R. (2008), Genetic algorithm for fuzzy logistics distribution vehicle routing problem, *Proceeding International Conference on Service Operations and Logistics, and Informatics*, 1427-1432, doi:10.1109/SOLI.2008.4686625.

Kuo, R.J., Chiu, C.Y. and Lin, Y.J. (2004), Integration of fuzzy theory and ant algorithm for vehicle routing problem with time window, *International Conference of the North American Fuzzy Information Processing Society*, **23**, 925-930.

Liu, C-S. and Lai, M-Y. (2009), The vehicle routing problem with uncertain demand at nodes, *Transportation Research Part E: Logistics and Transportation Review* **5(4)**, 517-524.

Lucic, P. and Teodorovic, D. (2003), Vehicle routing problem with uncertain demand at nodes: The Bee System and fuzzy logic Approach, In *Fuzzy Sets Based Heuristics for Optimization*, 67-82 Springer Verlag, Berlin.

Lucic, P. and Teodorovic, D. (2007), The fuzzy ant system for the vehicle routing problem when demand at nodes is uncertain, *Journal on Artificial Intelligence Tools* (IJAIT), **16(5)**, 751-770.

Resende, M.G.C. and Ribeiro. C.C. (2003), Greedy Randomized Adaptive Search Procedure. In F. Glover and G. Kochenberger (eds.), *Handbook in Metaheuristics*. Kluwer, pp. 219-249.

Stewart Jr., W.R. and Golden, B.L. (1983) Stochastic vehicle routing: a comprehensive approach, *European Journal of Operational Research*, **14(4)**, 371-385.

Tang, L., Cheng, W., Xhang, Z. and Zhong, B. (2007) Ant Colony Algorithm Based on Information Entropy Theory to Fuzzy Vehicle Routing Problem, *Proceedings ISKE*, series: *Advances in Intelligent Systems Research*.

Teodorovic, D. and Kikuchi, S. (1991), Application of fuzzy sets theory to the saving based vehicle routing algorithm, *Civil Engineering Systems*, **8**, 87-93.

Teodorovic, D. and Pavkovic, G. (1996), The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain, *Fuzzy Sets and Systems*, **82**, 307-317.

Teodorovic, D. (1999), Fuzzy Logic Systems for Transportation Engineering: The State of the Art. *Transportation Research*, **33(A)**, 337-364.

Tillman, F. (2002), The multiple terminal problem with probabilistic demands, *Transportation Science*, **3(3)**, 192-204.

Verdegay, J.L., Yager, R.R. and Bonissone, P. (2008), On heuristics as a fundamental constituent of Soft Computing, *Fuzzy Sets and Systems* **159**, 846-855.

Viertl, R. and Hareter, D. (2004) Fuzzy information and stochastics, *Iranian Journal of Fuzzy Systems*, **1 (1)**, 43-56.

Zhang, J.Y. and Li, J. (2007) Study on Logistics Distribution Vehicle Routing Problem with Fuzzy Due-Time, *International Conference on Management Science and Engineering*, 311-317, doi:10.1109/ICMSE.2007.4421866.

Zheng, Y.S. and Liu, B.D. (2006), Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm, *Applied Mathematics and Computation*, **176**, 673-683.

TWO MIXED INTEGER PROGRAMMING MODELS AND THEIR RELAXATIONS FOR THE MULTI-PORT MASTER BAY PLAN PROBLEM

Daniela Ambrosino[1], Davide Anghinolfi[2], Massimo Paolucci[2] and Anna Sciomachen[1]

[1]*Dep. of Economics, University of Genova, Italy, ambrosin@economia.unige.it, sciomach@economia.unige.it*

[2]*Dep. of, Computer Science, bioengineering, robotics and systems engineering (DIBRIS), University of Genova, Italy, paolucci@dist.unige.it, anghinolfi@dist.unige.it*

ABSTRACT

In this paper we extend the problem of determining how to stow a given set of containers of different types into the available locations of a containership, that is, the so-called Master Bay Plan Problem (MBPP), to the Multi-Port Master Bay Plan Problem (MP-MBPP). In the MP-MBPP the whole route of the ship is investigated; in particular, at each port of the route different sets of containers must be loaded for being shipped to next ports. Note that, differently from MBPP, in MP-MBPP at each port the sequence of two different handling operations affects the effectiveness of stowing plans: first, the import containers must be unloaded from the ship, then the export containers can be loaded.

We first introduce two exact mixed integer programming (MIP) formulations for the MP-MBPP; successively, we propose two heuristic approaches based on the solution of different relaxed formulations of the proposed MIP model. We report the computational tests performed on both random instances and real size ones which show the effectiveness of the models and the related proposed resolution methods.

Keywords: maritime logistics, stowage plans, mathematical programming, combinatorial optimization

1 INTRODUCTION

Maritime terminals play a strategic role within a logistic network, since many activities must be executed in order to optimize the flows of containers and minimize the total logistic cost. For this reason, a still increasing number of works have been recently proposed in the literature with the aim of improving the performances of both import /export and transhipment container terminals. A recent overview of relevant literature for these problems is provided in Stahlbock and Voss, (2008).

In this paper, we focus our analysis on the quay and ships activities; more precisely, we devote our attention to the stowage planning problem, with the aim of minimizing the overall berthing costs. Some works in the recent literature deal with the stowage of the containers on the ship, that is the so called master bay plan problem (MBPP). A detailed description of the problem together with its main constraints is given in Ambrosino et al. (2004). MBPP is a NP-Hard problem (Avriel et al. 2000). Heuristic methods for MBPP are compared in Ambrosino et al. (2010).

Note that the MBPP involves only the loading decisions at the first port without taking into account the possible loading operations at the next ports in the ship route. In this paper we extend the MBPP to the Multi-Port Master Bay Plan Problem (MP-MBPP), where a stowing plan is determined for each port in the route of the ship, so considering the sequence of ports that must be visited by the ship.

Since loading is one of the most difficult processes and the most affecting the efficiency of the terminal operations, we intend to optimize the loading and unloading operations at each port visited by the ship, considering all different sets of containers that must be loaded for being shipped to the next ports. Then, the present planning problem concerns the shipping line operator.

Few papers deal with the placement of containers into a containership on a multi-port journey. Imai et al. (2006) present a unified approach for taking into account the route planning problem from both the liner and the terminal manager point of view. A first model for dealing with the MP-MBPP has been presented by the authors in Ambrosino et al. (2009). In Wilson and Roach (2000) and Wilson et al. (2001) the authors propose a methodology for generating stowage plans for a containership on a multi-port journey that decomposes the decision process into two planning sub-processes: a strategic and a tactical one. The strategic objective function includes, among others, the minimization of the: i) number of cargo space occupied by each destination; ii) re-handles; iii) cargo blocks occupied by containers, and the maximization of the number of cranes used at each port. Zhang et al. (2005) decompose the stowage planning problem for a containership that serves many ports into two sub-problems as in Wilson and Roach (2000); their aim is to minimize re-handles and the number of bays occupied by the containers at each port.

In this paper, we propose and detail two mathematical formulations together with two solution approaches for facing the MP-MBPP at a strategic level (according to the decomposition in Wilson and Roach, 2000). Moreover, we report an extensive computational experimentation, performed on both random instances and real size ones, to evaluate the effectiveness of the proposed approach.

The remaining of the paper is organized as follows. In the next section we introduce the multi-port stowage planning problem in more details. In Section 3 we present the mathematical formulations for this NP-hard problem, while the solution approaches are described in Section 4. Results of our computational experimentation are reported in Section 5. Finally, in Section 6 we draw some conclusions and outline future works.

2 PROBLEM DEFINITION

The MP-MBPP consists in determining the stowage plan for a containership assigned to a specific route by the owner line operator in order to satisfy the service demand and to minimize the berthing time at each port, whilst satisfying the structural and operative constraints. More details about these constraints are reported in (Ambrosino et al., 2004). The route is made of a set of ports $P = \{1, ..., P_{max}\}$ that are visited in sequence by the ship. Here, we do not consider a circle route but a linear one. For each port $p \in P$ several sets of containers C_{pd} must be loaded in order to be shipped to ports d, $d \in P$ such that $d > p$. At each port p two sets of terminal operations are performed; in particular, first the containers bound for p are unloaded, so freeing ship stowage locations, then the containers departing from p to successive destinations (i.e., the demand at p) are loaded.

In this first attempt to solve the MP-MBPP, we introduce two simplifying assumptions that do not actually influence the applicability of the proposed approach: at the first port of the trip, i.e. $p = 1$, only loading operations are considered, whereas for $p = P_{max}$ only the relevant unloading operations are considered. Note that the ship is never empty after unloading at the last port of its route. The demand at port p can be further partitioned into several sets in accordance with the containers' type t ($t \in$ {20', 40'}), their class of weight g ($g \in$ {light, medium, heavy}) and their destination d, $d \in P$.

The unloading and loading operations at each port p are performed by a set of yard cranes which can work in parallel. We assume that the subsets of bays served by each crane at the different ports are a-priori defined and fixed. Of course the problems of defining both the number of quay cranes (QCs) to assign to a vessel and the quay crane schedule are strongly interrelated, also with deployment problems and stowage plans (see i.e. Bierwirth and Meisel 2010). Moreover, all these problems impact on the berthing time. In this analysis we solve the stowage plans by assuming to have a given number of QCs assigned to homogeneous parts of the ship, as this generally reflects the practise followed when unloading and loading operations are executed with the purposed of satisfying some operational and security constraints (i.e. weight balancing, spatial constraints, bays distance between cranes working a vessel, etc.). This assumption allows to minimize the berthing time by balancing the working time of quay cranes, as described in the following.

The MP-MBPP under investigation provides aggregate decisions for line planners that concern the assignment of containers to hatch hold/deck locations in order to satisfy the demand along the ship route. Each location of the ship is identified by means of its exact position with respect to its bay, tier and row address. As a consequence of the presence of hatches in the structure of the ship, it is natural to distinguish between hatch hold and hatch deck locations for properly defining stowage plans. Moreover, we assume that hatch locations on the deck or in the hold can be assigned to a single destination. Therefore, non-productive movements (re-handles) are required as soon as unloading and re-loading of containers in deck hatch locations are needed to access hold locations below the same hatch for unloading operations, as depicted in Figure 1. Note that containers in the deck locations are unloaded and then re-loaded not necessarily in the same hatch location. Of course, re-handles should be avoided as much as possible, as they increase both berthing time and handling costs.

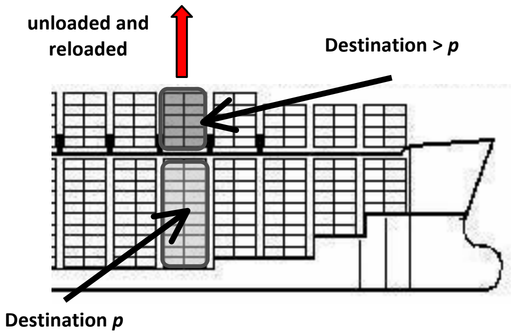


Figure 1. Non-productive movements (re-handles) at port *p*

Several objectives can be considered for the MP-MBPP. We model this multi-objective problem by introducing a scalar objective function including the following three components.

The first component penalizes the unfulfilled demand at the ports, that is, the situations in which some containers could not be loaded due to either location availability, or weight and stability constraints. The second important component is given by the penalization of re-handles. The last component aims at balancing the workload of the yard cranes available at a port. Since the loading and unloading operations at each port are performed in parallel by a set of yard cranes, the total operation time at a port corresponds to the longest one among the cranes; therefore, we penalize the difference between the operation times for each pair of cranes at the same port.

3 MATHEMATICAL MODELS FOR THE MP-MBPP

In this section we propose two mixed integer programming (MIP) model, denoted M1 and M2, for the MP-MBPP.

Let us fist summarize the required notation.

<u>Sets and indexes</u>

P set of ports (1 = first port, P_{max} = last port)
I set of bays of the ship ($I=E\cup O$, E set of even bays and O set of odd bays)
J set of rows of the ship
K set of tiers of the ship
H set of hatches of the ship
H_A / H_P subset of aft / prow hatches
H_R / H_L subset of right / left hatches
I_h subset of bays included in hatch h
R_h subset of rows included in hatch h
Y_p set of quay cranes serving the ship in parallel at port p
I_{pc} subset of bays served by crane c at port p
H_{pc} subset of hatches served by crane c at port p
G set of weight classes (e.g., 1=light, 2=medium, 3=heavy) of the containers
T set of types related to the size of containers (e.g. 20' and 40')

<u>Constants</u>

CH_{tij} / CD_{tij} number of hold /deck locations of type t available in bay i row j
$CAPH_h$ / $CAPD_h$ number of available locations (capacity) in the hold /deck of hatch h
$CAPH_{ht}$ / $CAPD_{ht}$ number of available locations of type t (capacity) in the hold /deck of hatch h
$TEUH_h$ /$TEUD_h$ capacity in TEU in the hold /deck of hatch h
N_{ptgd} number of containers of type t, weight class g to be loaded at port p with destination port d
W_g weight for a container in weight class g
Q_1 maximum cross equilibrium tolerance
Q_2 maximum horizontal equilibrium tolerance
PW_f penalization weight for the components of the objective function, f=1,...,3

<u>Variables</u>

$x\_h_{phtgd}\in\mathbf{Z}_+$, $x\_d_{phtgd}\in\mathbf{Z}_+$ $\forall p\in P, h\in H, t\in T, g\in G, d\in P\ d>p$
Hatch hold/deck state variables; x_h_{phtgd} (x_d_{phtgd}) is the number of containers of type t, weight class g with destination port d located in the hold (on the deck) below (above) hatch h after loading at port p.

$l\_h_{phtgd}\in\mathbf{Z}_+$, $l\_d_{phtgd}\in\mathbf{Z}_+$ $\forall p\in P, h\in H, t\in T, g\in G, d\in P\ d>p$
Hatch loading variables; l_h_{phtgd} (l_d_{phtgd}) is the number of containers of type t, weight class g with destination port d loaded in the hold (on the deck) below (above) hatch h at port p.

$y\_h_{phd}\in\{0,1\}$, $y\_d_{phd}\in\{0,1\}$ $\forall p\in P, h\in H, d\in P, d>p$
Hatch assignment variables for hold/deck locations; y_h_{phd}=1 if at port p the locations in hold (on deck) below (above) hatch h are assigned to containers bound for d.

$u\_h_{pij}\in\{0,1\}$, $u\_d_{pij}\in\{0,1\}$ $\forall p\in P, i\in I, j\in J$
Use of bay-row; u_h_{pij}=1 (u_d_{pij}=1) if at port p bay i and row j in the hold (deck) is used. Note that if i is even then 40' containers can be loaded, otherwise 20' ones can be loaded.

$r_{phtgd}\in\mathbf{Z}_+$ $\forall p\in P, h\in H, t\in T, g\in G, d\in P\ d>p$
Non-productive movement variables; r_{phtgd} is the number of re-handles executed at port p for unloading (and successively reloading) the containers bound for $d>p$ over hatch h on the deck for unloading containers in the hold below h bound for p.

$f_{ph}\in\{0,1\}$ $\forall p\in P, h\in H$

Need to free deck locations over hatch (non-productive movements); f_{ph}=1 if the containers on the deck over hatch h must be removed at port p to unload other containers from the hold below h.

$v_{ptgd} \in \mathbf{Z}_+ \ \forall p \in P, t \in T, g \in G, d \in P, d > p$
Number of not served containers; v_{ptgd} represents the demand not satisfied at port p, that is the number of containers of type t, weight class g with destination port d not loaded at port p.

$\Delta_{pce} \in \boldsymbol{R} \ \forall p, \forall c, e \in Y_p, c \neq e$
Absolute difference between the number of unloading and loading operations executed by cranes c and e at port p.

$t_{pc} \in \boldsymbol{R} \ \forall p \in P, c \in Y_p$
Total number of loading and unloading operations of crane c at port p.

$\Delta max \in \boldsymbol{R}$
Maximum difference in the number of handling operations between each pair of cranes over all the considered ports.

The three objective function components described in the previous section can be then expressed as:

- $U(v) = \sum_{ptgd} v_{ptgd}$ number of not served containers;
- $NP(r) = \sum_{phtgd} r_{phtgd}$ number of non-productive movements (i.e. re-handles).
- $MTV(\Delta) = \Delta max$ maximum absolute variation between the number of operations performed on the ship by pairs of cranes at the ports (i.e., crane unbalance);

3.1 The First Model: Model M1

The first proposed Multi-Port MBPP MIP formulation (M1) is then the following:

$$\min PW_1 \cdot U(v) + PW_2 \cdot NP(r) + PW_3 \cdot MTV(\Delta) \tag{1}$$

subject to

$$\sum_h (l\_h_{phtgd} + l\_d_{phtgd}) = N_{ptgd} + \sum_h r_{phtgd} - v_{ptgd} \quad \forall p, t, g, d \tag{2}$$

$$x\_h_{phtgd} = x\_h_{p-1htgd} + l\_h_{phtgd} \quad \forall p, h, t, g, d \tag{3}$$

$$x\_d_{phtgd} = x\_d_{p-1htgd} - r_{phtgd} + l\_d_{phtgd} \quad \forall p, h, t, g, d \tag{4}$$

$$\sum_{tgd} l\_h_{phtgd} \le CAPH_h \cdot (1 - \sum_{d>p} y\_d_{p-1hd}) \quad \forall 1 < p < P, h \tag{5}$$

$$\sum_g x\_h_{phtgd} \le CAPH_{ht} \cdot y\_h_{phd} \quad \forall p, h, t, d \tag{6}$$

$$\sum_g x\_d_{phtgd} \le CAPD_{ht} \cdot y\_d_{phd} \quad \forall p, h, t, d \tag{7}$$

$$\sum_d y\_h_{phd} \le 1 \ \ \forall p, h \tag{8}$$

$$\sum_d y\_d_{phd} \le 1 \ \ \forall p, h \tag{9}$$

$$y\_h_{phd} \ge y\_h_{p-1,hd} \quad \forall p, h, d \, (d>p) \tag{10}$$

$$y\_d_{phd} \ge y\_d_{p-1,hd} - f_{ph} \quad \forall p, h, d \, (d>p) \tag{11}$$

$$y\_h_{ph,p+1} + y\_d_{phd} \le 1 + f_{p+1,h} \quad \forall h, \forall p, d > p+1 \tag{12}$$

$$r_{phtgd} \le CAPD_{ht} \cdot f_{ph} \quad \forall p, h, t, g, d \tag{13}$$

$$r_{phtgd} \ge x\_d_{p-1,tgd} - CAPD_{ht} \cdot (1 - f_{ph}) \quad \forall p, h, t, g, d \tag{14}$$

$$\sum_g x\_h_{ph20gd} + 2 \cdot \sum_g x\_h_{ph40gd} \le TEUH_h \cdot y\_h_{phd} \quad \forall p, h, d > p \tag{15}$$

$$\sum_{g} x\_d_{ph20gd} + 2 \cdot \sum_{g} x\_d_{ph40gd} \leq TEUD_h \cdot y\_d_{phd} \quad \forall p,h,d > p \tag{16}$$

$$u\_d_{pij} + u\_d_{pi+1j} = 1 \ \forall p, i \in E, j \in J \tag{17}$$

$$u\_d_{pij} + u\_d_{pi-1j} = 1 \ \forall p, i \in E, j \in J \tag{18}$$

$$u\_h_{pij} + u\_h_{pi+1j} = 1 \ \forall p, i \in E, j \in J \tag{19}$$

$$u\_h_{pij} + u\_h_{pi-1j} = 1 \ \forall p, i \in E, j \in J \tag{20}$$

$$\sum_{gd} x\_h_{phtgd} \leq \sum_{i \in I_h} \sum_{j \in R_h} CH_{tij} \cdot u\_h_{pij} \quad \forall p,h,t \tag{21}$$

$$\sum_{gd} x\_d_{phtgd} \leq \sum_{i \in I_h} \sum_{j \in R_h} CD_{tij} \cdot u\_d_{pij} \quad \forall p,h,t \tag{22}$$

$$\left| \sum_{h \in H_L, tgd} W_g (x\_h_{phtgd} + x\_d_{phtgd}) - \sum_{h \in H_R, tgd} W_g (x\_h_{phtgd} + x\_d_{phtgd}) \right| \leq Q_1 \quad \forall p \in P \tag{23}$$

$$\left| \sum_{h \in H_A, tgd} W_g (x\_h_{phtgd} + x\_d_{phtgd}) - \sum_{h \in H_P, tgd} W_g (x\_h_{phtgd} + x\_d_{phtgd}) \right| \leq Q_2 \quad \forall p \in P \tag{24}$$

$$t_{pc} = \sum_{h \in H_{pc}} \sum_{tgd} (l\_h_{phtgd} + l\_d_{phtgd} + r_{phtgd}) + \sum_{h \in H_{pc}} \sum_{tgd} (x\_h_{p-1,htgp} + x\_d_{p-1,htgp}) \quad \forall p, c \in Y_c \tag{25}$$

$$| t_{pc} - t_{pe} | \leq \Delta_{pce} \quad \forall p \in P, \forall c, e \in Y_p, c \neq e \tag{26}$$

$$\Delta max \geq \Delta_{pce} \quad \forall p \in P, \forall c, e \in Y_p, c \neq e \tag{27}$$

$$\sum_{gd} x\_h_{ph20gd} \geq \sum_{i \in I_h \cap E} CH_{20ij} \cdot u\_h_{pij} \ \forall p,h : \exists CH_{20ij} > 0 \ i \in I_h \cap E \tag{28}$$

$$
\begin{array}{ll}
x\_h_{phtgd} \in \mathbf{Z}_+ , x\_d_{phtgd} \in \mathbf{Z}_+ & \forall p, h, t, g, d, d > p \\
l\_h_{phtgd} \in \mathbf{R}_+ , l\_d_{phtgd} \in \mathbf{R}_+ & \forall p, h, t, g, d, d > p \\
r_{phtgd} \in \mathbf{R}_+ & \forall p, h, t, g, d, d > p \\
y\_h_{phd} \in \{0,1\}, y\_d_{phd} \in \{0,1\} & \forall p, h, d, d > p \\
f_{ph} \in \{0,1\} & \forall p, h \\
v_{pgd} \in \mathbf{R}_+ & \forall p, g, d, d > p \\
u\_h_{pij} \in \{0,1\}, u\_d_{pij} \in \{0,1\} & \forall p, i, j \\
t_{pc} \in \mathbf{R}_+ & \forall p, c \in Y_p \\
\Delta_{pce} \in \mathbf{R}_+ & \forall p, \forall c,e \in Y_p, c \neq e \\
\Delta max \in \mathbf{R}_+ &
\end{array}
\tag{29}
$$

Equation (1) is the objective function. Constraints (2) are the demand constraints and define, for each port and each subset of containers, the number of loaded and unloaded containers. Note that, for each port, the number of containers to be loaded also includes the unloaded containers due to re-handles. Constraints (3) and (4) represent the state continuity conditions, asking that the number of homogeneous containers (i.e. having the same type, class of weight and destination) located into a hatch at port *p* is updated taking into account the containers already located there at port *p*-1 and the containers loaded at *p*; for the deck locations in (4) the containers possibly removed from those locations due to re-handles are considered too. Constraints (5) are related to the hold accessibility; in particular, such constraints enable stowing containers in the hold locations of hatch *h* (according to the destination assigned to the hatch), only if the deck above *h* is free.

Constraints (6) and (7) link variables y_h_{phd} and y_d_{phd}, used to assign a destination to hold/deck locations of hatch *h*, to hold/deck state variables, respectively; thanks to these capacity constraints at port *p* it is not possible to have into hold/deck locations of hatch *h* assigned to destination *d* containers with a different destination; some *x_h* and *x_d* variables are then forced to 0. Constraints (8) and (9) impose that hold and deck locations corresponding to a hatch can be assigned only to one destination.

Constraints (10) impose that hold locations of hatch *h* assigned to destination *d* remain assigned to *d* until port *d* is reached; differently, due to constraints (11), locations on the deck can be freed before destination if some re-handles are required.

Constraints (12) fix $f_{ph} = 1$ when, at port *p*, it is necessary to remove the containers on the deck over hatch *h* to unload other containers from the hold below *h*. Constraints (13) guarantee that containers are removed (re-handles) from deck locations in hatch *h*, only if hatch *h* must be freed at *p* (i.e., $f_{p,h}$=1); in this case, constraints (14) define the exact number of re-handles, that is the number of containers to be unloaded from the deck locations.

Constraints (15) and (16) are TEU capacity conditions and impose that the sum of TEUs located in the hold of a hatch/deck for the assigned destination must not exceed the maximum available TEU capacity for that hatch/deck (note that $TEUH_h = CAPH_{20h}$). Constraints (17)-(20) are related to the usage of the locations of each bay. The even bay (*i*) can be used for 40' containers, otherwise the corresponding odd bays (*i*-1, *i*+1) can be used for 20' containers; the decision is separated for hold and deck locations. Constraints (21) and (22) are the capacity constraints that take into account the bay usage, as defined by constraints (17)-(20). The stack capacity changes according to the bay usage. Containers of 40' cannot be stored in odd bays, that is C_{20ij} is equal to 0 for index *i* corresponding to even bays.

Constraints (23) and (24) represent the stability conditions that are related to the horizontal and cross equilibrium. Constraints (25)-(27) are related to the cranes utilization. In particular, constraints (25) define variable t_{pc} that is the total number of loading and unloading operations for each crane at each port. Constraints (26) and (27) define, respectively the absolute difference between the number of unloading and loading operations at port *p* needed by cranes *c* and *e* and the maximum difference of loading and unloading operations between any pair of cranes over all the considered ports.

When dealing with real ship planning problems there is the possibility of having irregular keels; in these cases particular stowage are needed. For instance, a container of 20' is necessary for granting the stability of the stack of 40' containers, as showed in Figure 2. If it is the case, constraints (28) prevent possible inconsistencies in loading even bays in the hold in those rows including single 20' locations under 40' ones. Such constraints force the loading of 20' containers when the bay-row is assigned to 40' containers, thus avoiding that any 40' container is positioned over an empty 20' location. Finally, (29) define the decision variables.

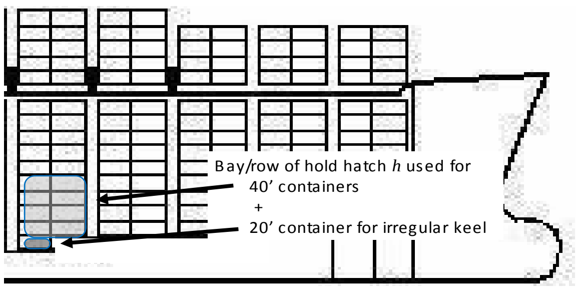


Figure 2. Example of stowage of containers into irregular keel

3.2 Model M1 Versus Model M2

In this section we briefly describe a different formulation (M2) for the MP-MBPP that we obtain by relaxing model M1; in particular, in M2 we consider assignments, loading operations and re-handles in terms of TEUs, instead of containers as in M1. As a first consequence, variables u_h_{pij} and u_d_{pij}, that in M1 establish the exact usage of locations in even and odd bays for 20' and 40' containers, respectively, in M2 are not anymore necessary; the same is for constraints related to the bays usage (17)-(20), and to the capacity of the ship that, as stated in (21) and (22), depends on the bays usage. Moreover, all variables of M1 related to the number of containers of a given type are replaced in M2 by similar variables that specify the number of TEUs. Note that, in this way, in M2 formulation we no longer use index t in hatch state, loading and re-handling variables. Just to give an example, we use variables x_h_{phgd} and x_d_{phgd} in M2 to represent the number of TEUs of containers of weight class g, bound for d, assigned to hatch h departing from port p instead of the corresponding variables x_h_{phtgd} and x_d_{phtgd} in M1. In the same way, demand satisfaction constraints (2) are now expressed in terms of TEUs instead of number of containers of type t, and hence constant N_{pgd} is now used without considering index t. Analogously, constraints (6), (7), (13) and (14) are formulated in M2 considering constants $TEUH_h$ and $TEUD_h$ to account the TEU capacities of hatches instead of constants $CAPH_{ht}$ and $CAPD_{ht}$ in M1. Furthermore, constraints similar to those used in M1 to model state continuity (3) and (4), stability conditions (23)-(24) and crane utilization (25) are expressed in M2 in terms of the new TEU-based variables. Finally, in M2 the TEU capacity constraints (15) and (16) are disregarded as they reduce to (6) and (7) using the new TEU-based state variables.

4 SOLVING THE MP-MBPP FROM RELAXATIONS OF M1 AND M2

Since model M1 can be solved up to optimality only for small instances, we develop two different heuristic approaches denoted, respectively, HM1 and HM2M1, based on the solution of relaxations of models M1 and M2. In both cases, the idea is to face MP-MBPP in two steps, decomposing the original complex planning problem into two simpler ones that are sequentially solved. Heuristic HM1 is based on model M1. More precisely, in the first step of HM1 we consider a continuous relaxation of model M1, in which all general integer variables are relaxed to assume real values, while binary variables remain constrained to be 0/1 for determining the assignment of destinations to hatch locations. In the second step of HM1, we get the optimal solution of the first step and solve the original model M1, fixing the binary assignment variables. Note that the solution obtained in the second step is a feasible solution for the MP-MBPP. Heuristic HM1 is summarized in Figure 3a.

Heuristic approach based on formulation M1 – HM1	*Heuristic approach based on formulations M2 and M1 – HM2M1*
Step 1 Solve a Partially Relaxed M1 (PR_M1) use the continuous relaxation of the following variables and solve hatch state (x_h_{phtgd}, x_d_{phtgd}) load (l_h_{phtgd}, l_d_{phtgd}) re-handles (r_{phtgd}) not loaded containers (v_{ptgd}) Step 2 Solve a Partially Fixed M1 (PF_M1) fix the following binary variables as in the solution of PR_M1 and solve hatch assignment (y_h_{phd}, y_d_{phd}) bay/row usage (u_h_{pij}, u_d_{pij})	Step 1 Solve a Partially Relaxed TEU model (PR_M2) use the continuous relaxation of the following variables and solve hatch state (x_h_{phgd}, x_d_{phgd}) load (l_h_{phgd}, l_d_{phgd}), re-handles (r_{phgd}) not loaded containers (v_{pgd}) Step 2 Solve a Partially Fixed M1 (PF_M1) fix the following binary variables as in the solution of PR_M2 and solve hatch assignment (y_h_{phd} , y_d_{phd})

Figure 3a. Steps of heuristic HM1 **Figure 3b**. Steps of heuristic HM2M1

As it will be shown in more details in the next section, the computational tests performed with heuristic HM1 provide appreciable results; however, the first step of HM1 requires a high computational effort that could be impracticable for large size instances. Thus, we devise heuristic HM2M1, based on both model M1 and M2. In the first step we relax all general integer variables and solve model M2 with only continuous and binary variables. In the second step of HM2M1, as it has been done in the case of H1, we solve model M1, fixing the binary variables to the optimal values obtained in the first step. Heuristic HM2M1 is summarized in Figure 3b.

5 EXPERIMENTAL RESULTS

In this section we report the main results obtained through an extensive computational experimentation, performed on both real size instances and random generated ones of the MP-MBPP. In particular, we define two sets of 18 instances for the following containerships:

- the "European Senator" (ES), that is a medium sized ship client of the terminal SECH, located in Genova, Italy, with capacity of 2028 TEUs, including 17 odd bays, 10 rows and 6 tiers in the hold and 21 odd bays, 12 rows and 5 tiers in the upper deck, 33 hatches;
- a fictitious "Large Ship" (LS), with capacity of 5280 TEUs, including 40 odd bays, 12 rows and 6 tiers in the hold and 40 odd bays, 12 rows and 5 tiers in the upper deck, 60 hatches.

In both cases we consider a route of 7 ports, in which loading operations are executed at ports 1-5, whilst unloading operations are executed at ports 2-7, being the ship empty at port 1. We also assume that at each port either 2 or 4 cranes are used for ES and LS, respectively.

The main characteristics of the tested instances are related to: *i) the load level*: the considered instances have been partitioned into three subsets according to the load level (low, medium and high) represented by both the occupancy of the ship and the average TEUs to load at each port of the trip. Occupancy (i.e. an average percentage of loading with respect to the ship capacity) is 50%, 70% and 85% respectively for low, medium and high load level instances*; ii) the load type*: in each subset we fix two levels of percentage of 20' containers, 50 - 70 %.

We implemented models M1 and M2 in Ilog OplStudio 5.2, with MIP solver Cplex 11.0.1, and ran all tests on a 2.4GHz Intel Core 2 Duo E6600 computer with 4GB RAM. We executed the tests for model M1 by setting a time limit bound of 1800s, for ES instances, and 3600s for LS ones. Model M1 requires 30829 variables (including 7104 binaries and 11088 integers) and 36096 constraints for ES instances, and 56732 variables (including 13680 binaries and 20160 integers) and 66071 constraints for LS instances.

The three components in the objective function are homogeneous since they report the number of containers not loaded, re-handled and moved by cranes. By a lexicographic approach we give an higher weight to the first component of the objective function that represents the main purpose of planning (i.e., to serve as much as possible the transportation demand); then, we penalise unproductive movements when unloading/loading the ship and finally, we try to obtain the best performance in terms of minimization of berthing time by minimizing the crane work unbalance that is the difference in the number of operations executed by each crane, since we assumed that cranes are devoted to serve homogeneous portions of the ship. Following this approach the chosen weights of the objective function are: PW3=100, PW2=10, PW1=1.

The MIP solver was not able to solve any instance of the large ship (LS) within the chosen time limit, whereas, as showed in Table 1, some instances of the medium ship (ES) can be solved up to optimality within 1800s.

Table 1. Results obtained by solving instances ES by model M1

Inst	Ship - load	Obj	Time (limit=1800s)	Not served	Re-handles	Crane Unbalance
1	ES –low - 50	1	952	0	0	1
2	ES –low - 50	0	530	0	0	0
3	ES –low - 50	1	1109	0	0	1
4	ES –low - 70	1	677	0	0	1
5	ES –low - 70	1	449	0	0	1
6	ES –low - 70	1	594	0	0	1
7	ES –medium - 50	no	1800	0	0	0
8	ES –medium - 50	6245	1800	52	101	35
9	ES –medium - 50	no	1800			
10	ES –medium - 70	1	1539	0	0	1
11	ES –medium - 70	1	1800	0	0	1
12	ES –medium - 70	11	1800	0	1	1
13	ES –high - 50	no	1800			
14	ES –high – 50	no	1800			
15	ES –high - 50	no	1800			
16	ES –high - 70	no	1800			
17	ES –high - 70	no	1800			
18	ES –high - 70	no	1800			

Tables 2 and 3 show the results related to ES and LS, respectively, obtained with heuristic HM1. We can observe that we are able to get quite good results within the given time limit for ES instances, whereas the results are not so positive for LS instances.

In particular, readers can note that many containers are not served; more precisely, the number of not served containers ranges from 113, in instance 17, up to 204, in instance 15. Furthermore, in case of instances 10, 13-18 the relaxed version of model M1, used in step 1, cannot be solved up to optimality within 1800s and a large number of re-handles (on the average 164) is needed. For this reason, in the last six rows of Table 3, we add the results obtained by extending the time limit to 3600s for the subset of the most difficult LS instances.

Table 2. Results for (ES) instances obtained by heuristic HM1

Inst	PR-M1 obj	CPU	Not served	Re-handles	Crane unbalance	PF-M1 obj	CPU	Not served	Re-handles	Crane unbalance
1	0	308	0	0	0	11	7	0	0	11
2	0	342	0	0	0	0	5	0	0	0
3	0	300	0	0	0	1	7	0	0	1
4	0	393	0	0	0	1	5	0	0	1
5	0	293	0	0	0	1	7	0	0	1
6	0	244	0	0	0	1	6	0	0	1
7	0	981	0	0	0	65	6	0	0	65
8	0	605	0	0	0	40	6	0	0	40
9	0	825	0	0	0	2	5	0	0	2
10	0	511	0	0	0	1	6	0	0	1
11	0	547	0	0	0	97	6	0	0	97
12	0	523	0	0	0	14	5	0	0	14
13	810	1800	8	1	0	1020	7	8	1	210
14	30	1800	0	3	0	52	7	0	3	22
15	60	1800	0	6	0	116	6	0	0	116
16	807.83	1800	2	58	27.83	872	7	0	75	122
17	0	1263	0	0	0	77	7	0	0	77
18	500	1800	5	0	0	819	6	5	0	319

Table 3. Results for (LS) instances obtained by heuristic HM1

Inst	PR-M1 obj	CPU	Not served	Re-handles	Crane unbalance	PF-M1 obj	CPU	Not served	Re-handles	Crane unbalance
1	0	372	0	0	0	1	18	0	0	1
2	0	789	0	0	0	1	20	0	0	1
3	0	355	0	0	0	1	18	0	0	1
4	0	465	0	0	0	2	19	0	0	2
5	0	412	0	0	0	1	21	0	0	1
6	0	518	0	0	0	1	20	0	0	1
7	0	1305	0	0	0	1	17	0	0	1
8	0	974	0	0	0	1	25	0	0	1
9	0	1058	0	0	0	2	30	0	0	2
10	20838.19	1800	167	411	28.19	18346	33	146	369	56
11	0	692	0	0	0	1	21	0	0	1
12	0	862	0	0	0	1	18	0	0	1
13	20735.50	1800	175	319	45.50	19790	16	164	334	50
14	22535.50	1800	189	345	185.50	20285	17	161	402	165
15	23907.73	1800	216	230	7.73	23220	17	204	281	10
16	24832	1800	226	220	32	20675	23	170	365	25
17	20430.20	1800	168	352	110.20	15760	35	113	437	90
18	23499	1800	201	330	99	22790	20	191	360	90
10	0	2023	0	0	0	1	17	0	0	1
13	102.33	3600	1	0	2.33	103	20	1	0	3
15	542.67	3600	0	52	22.67	543	19	0	52	23
16	6060	3600	56	46	0.00	2923	20	20	92	3
17	17650	3600	152	237	80.00	14931	27	115	343	1
18	2584.50	3600	14	116	24.50	1883	22	4	146	23

Table 4. The results for (ES) instances obtained by heuristic HM2M1

Inst	PR-M2 obj	CPU	Not served	Re-handles	Crane unbalance	PF-M1 obj	CPU	Not served	Re-handles	Crane unbalance
1	0	17	0	0	0	1	36	0	0	1
2	0	19	0	0	0	0	20	0	0	0
3	0	8	0	0	0	1	18	0	0	1
4	0	7	0	0	0	1	578	0	0	1
5	0	15	0	0	0	2	14	0	0	2
6	0	13	0	0	0	32	11	0	3	2
7	0	24	0	0	0	1	557	0	0	1
8	0	59	0	0	0	1	94	0	0	1
9	0	12	0	0	0	2	88	0	0	2
10	0	27	0	0	0	1	28	0	0	1
11	0	44	0	0	0	1	70	0	0	1
12	0	32	0	0	0	1	27	0	0	1
13	0	84	0	0	0	1	119	0	0	1
14	0	66	0	0	0	1	251	0	0	1
15	0	81	0	0	0	1	45	0	0	1
16	0	132	0	0	0	2	43	0	0	2
17	0	69	0	0	0	1	83	0	0	1
18	0	78	0	0	0	1	57	0	0	1

Tables 4 and 5 show the results obtained with heuristic HM2M1. Model M2, solved at step1 of HM2M1 with a time limit of 900s, has 13857 variables (including 1980 binaries) and 18275 constraints for ES instances, while 25165 variables (including 3600 binaries) and 33244 constraints are required for LS instances.

Observing Tables 4 and 5, we must emphasize that HM2M1 is very fast in both phases; in fact final solutions are obtained on the average in 162s for ES instances and in 393s for LS ones. In addition, the results obtained by HM2M1 in terms of the objective function (1) appear outstanding for the following reasons: (a) such results clearly dominates the ones produced by HM1; (b) the absolute gaps from the lower bounds of M1, which are always obviously zero for all the instances, are very reduced since, apart from instance 6 with 3 re-handles, they correspond at most to a crane unbalance of 2 operations.

Table 5. The results for (LS) instances obtained by heuristic HM2M1

Inst	PR-M2 obj	CPU	Not served	Re-handles	Crane unbalance	PF-M1 obj	CPU	Not served	Re-handles	Crane unbalance
1	0	33	0	0	0	1	65	0	0	1
2	0	51	0	0	0	2	149	0	0	2
3	0	70	0	0	0	2	32	0	0	2
4	0	39	0	0	0	2	105	0	0	2
5	0	41	0	0	0	2	161	0	0	2
6	0	54	0	0	0	1	114	0	0	1
7	0	136	0	0	0	1	238	0	0	1
8	0	137	0	0	0	1	192	0	0	1
9	0	117	0	0	0	1	188	0	0	1
10	0	163	0	0	0	1	108	0	0	1
11	0	120	0	0	0	1	174	0	0	1
12	0	94	0	0	0	1	95	0	0	1
13	0	258	0	0	0	1	163	0	0	1
14	0	220	0	0	0	2	653	0	0	2
15	0	238	0	0	0	1	671	0	0	1
16	0	241	0	0	0	1	277	0	0	1
17	0	244	0	0	0	1	845	0	0	1
18	0	270	0	0	0	1	291	0	0	1

In order to better compare the performances of the proposed heuristics, in Table 6 we summarize the related results. In particular, column *Total CPU time* shows the percentage of CPU time savings when applying HM2M1 instead of HM1; columns *Deviations* report the differences between not loaded, re-handles and crane unbalance results obtained by HM1 and HM2M1; in particular, positive numbers denote that HM2M1 outperforms HM1. From Table 6 we can observe that HM2M1 is less computational time consuming than HM1; moreover, HM2M1 is able to allocate all the demand for both sets of instances without re-handles, apart from instance 6 of ES. Furthermore, for all instances the workload unbalance of the bay cranes obtained by HM2M1 is negligible. Finally, HM2M1 obtains the optimal solution in 5 over 7 instances (i.e., instances 1-4, and 7), that have been solved up to optimality by M1 (see Table 1).

Table 6. Performance's comparison of the proposed solution approaches

Inst	ES				LS			
	Total CPU time (%)	Deviations			Total CPU time %	Deviations		
		Not-served	Re-handles	Crane unbalance		Not-served	Re-handles	Crane unbalance
1	83.17	0	0	10	74.87	0	0	0
2	88.76	0	0	0	75.28	0	0	-1
3	91.53	0	0	0	72.65	0	0	-1
4	-46.98	0	0	0	70.25	0	0	0
5	90.33	0	0	-1	53.35	0	0	-1
6	90.40	0	-3	-1	68.77	0	0	0
7	41.13	0	0	64	71.71	0	0	0
8	74.96	0	0	39	67.07	0	0	0
9	87.95	0	0	0	71.97	0	0	1
10	89.36	0	0	0	86.72	0	0	0
11	79.39	0	0	96	58.77	0	0	0
12	88.83	0	0	13	78.52	0	0	0

13	88.77	8	1	209	88.37	1	0	2
14	82.46	0	3	21	51.95	161	402	163
15	93.02	0	0	115	74.88	0	52	22
16	90.32	0	75	120	85.69	20	92	2
17	88.03	0	0	76	69.98	115	343	0
18	92.52	5	0	318	84.51	4	146	22

6 CONCLUSIONS

In this paper we deal with the multi-port stowage planning problem, denoted MP-MBPP. It is a complex planning problem whose solution can support line operators in optimizing the use of scarce resources (ship locations), redefining aggregate stowage plans reacting to last minute changes in the transportation demand.

At least to the authors' knowledge, there are not in the recent literature detailed mathematical formulations and solution approaches able to faces this NP-hard combinatorial optimization problem. We propose an exact MIP formulation and a relaxed one, based on TEU capacity constraints; then, we develop two different solution methods that solve the problem using two-step decomposition approach. We tested the proposed heuristics on both real life instances and fictitious ones. The obtained results show that the second heuristic is able to define nearly optimal stowage plans for 5600 TEU containership on a 7 port route in a quite reduced computation time. In a future extension of this research we will analyse how to extend the proposed approach for dealing with larger instances. In addition, a rolling horizon procedure can be developed in order to face planning problems including circular routes.

REFERENCES

Ambrosino, D., Anghinolfi, D., Paolucci, M., Sciomachen, A. (2009), Modelling the multi-port master bay plan problem. In: R.R-M Dauer; R.M. Sagarra; F.X. Martinez De Oss. *Maritime Transport*. vol. iv, p. 63-73, Barcellona.

Ambrosino, D., Anghinolfi, D., Paolucci, M., Sciomachen, A. (2010), An experimental comparison of different metaheuristics for the Master Bay Plan Problem. In: P. Festa. *Experimental Algorithms*, Lecture Notes in computer Science. p. 314-325, Springer.

Ambrosino, D., Sciomachen, A. and Tanfani, E. (2004), Stowing a containership: The Master Bay Plan problem, *Transportation Research A*, **38**, 81-99.

Avriel, M., Penn, M. and Shpirer N. (2000), Container ship stowage problem: complexity and connection to the colouring of circle graphs, *Discrete Applied Mathematics*, **103**: 271-279.

Bierwirth C., Meisel F. (2010). A survey of berth allocation and quay crane scheduling problem in container terminals. *European Journal of Operational Research*, 202 (3), 615-627.

Imai, A., Sasaki, K., Nishimura, E., Papadimitriou, S. (2006), Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks, *European Journal of Operational Research*, **171**, 373–389.

Stahlbock, R. and Voss, S. (2008), Operations research at container terminal: a literature update, *OR Spectrum*, **30**, 1-52.

Wilson, I.D., Roach, P.A. (2000), Container stowage planning: a methodology for generating computerised solutions, *Journal of the Operational Research Society*, **51**,11, 1248-1255.

Wilson, D, Roach, P. A. and Ware, J. A (2001), Container stowage pre-planning: using search to generate solutions, a case study, *Knowledge-Based Systems*, ***14**, 3-4, 13*7-145.

Zhang, W-Y, Lin, Y., Zhou-Shang, JI (2005), Model and algorithm for container ship stowage planning based on bin packing problem, *Journal of Marine Science and application*, **4**(3), 30-36.

CAPACITY EVALUATION OF INLAND CONTAINER TERMINALS - THE SIMULATION BASED APPROACH OF SIMCONT

Manfred Gronalt[1], Martin Posset[2] and Soraya Rojas-Navas[2]

[1]*University of Natural Resources and Life Sciences, Vienna, Feistmantelstraße 4 1180 Vienna, manfred.gronalt@boku.ac.at*

[2]*H2 Projektberatung KG, Austria, mcp@h2pro.at*

ABSTRACT

In this paper we show the SimConT approach in simulation of inland container terminals. SimConT was developed and used during the last years in several projects on capacity evaluation of inland container terminals. We present the structure of the SimConT approach and modules and show how we apply it in real life applications. The first step is collection of infrastructure data, train related and load carrier data. Further, the detailed operation of the yard hat to be modelled and finally during the analysis phase the study has to proceed stepwise to reach the marginal productivity rate of the Container Terminal. For this we are using a scenario tree in order to move from simulation result to the next scenario in order to generate a feasible capacity expansion line. We show the results from a real life application to demonstrate our approach.

Keywords: inland container terminal, rail-rail and rail-truck terminal, simulation study approach

1 INTRODUCTION

Inland container terminals (CT) hold a crucial role in modern supply chains, as they are the essential transshipment points for containerized freight between transport modes by land and sea, and function as feeder terminals for open sea terminals. Given the increasing importance of inland terminals and increasing flows of goods, rail transport must be strengthened as favoured mode of transport. Franc and van der Horst (2010) show that shipping lines and terminal operators are broaden their scope in business to hinterland and inland terminals in order to prepare for future volatile supply chains and to improve empty container supply.

Over the last years a simulation tool called SimConT was developed which is tailored to the special characteristics and requirements of inland container terminals. The tool allows for strategic and tactical simulation of terminal infrastructure and operations. By means of simulation, decision makers are actively supported in planning processes, while minimizing the risk of bad investments and stranded costs when planning and (re)building terminal infrastructure and enlarge terminal capacity. Inland CT mark the peripheral nodes in efficient freight transport. They distribute and collect containers and other intermodal load carriers like swap bodies and ‘liftable’ semi trailers. Inland CT receive and ship containers from sea terminals, another inland CT or from local industries. In inland CT Intermodal Transportation Units (ITE) usually represent a mix of the above mentioned load carriers. We can observe that in the recent past the portion of liftable trailers was steadily increasing. This is an effect of increasing intra European freight transport modus shift from road to rail. All over Europe approx. 800 CT of different sizes are supporting the intermodal transportation network. An ongoing change in European transport infrastructure also leads to requirements to adapt CT to new tasks.

Figure 1. Inland CT Layout

In Fig.1 a typical CT Layout is shown. We can see two cranes and three reach stackers, serving the storages areas. The gates for the trucks and the truck waiting area are arranged in the eastern part of the terminal.

We will now define some basic principles on how to model inland CT. One key element in the operation of these inland CT is the organization of ITE – exchange. We denote this as operation strategies (Benna and Gronalt, 2008) for inland CTs. The second element which contributes to the performance of inland CT is infrastructure: rail yard tracks, cranes, stacker for single or multi terminal environment and terminal infrastructure network for movement of stackers, trucks, terminal tractors and storage capacity and portion of dedicated storage areas of the CT. Further, we have to consider the arrival pattern of trains and trucks, the ITE-mix on trains and the relation between import and export ITE as order systems of the CT. Fig. 2 shows how these elements are used as a standard input for SimConT Simulation module in order to analyze CT performance. The configuration module collects all relevant data and prepares the terminal parameters for the simulation module. The simulation module generates CT orders, train and truck arrival patterns and calculates CT performance data, which are passed on to a report generator.

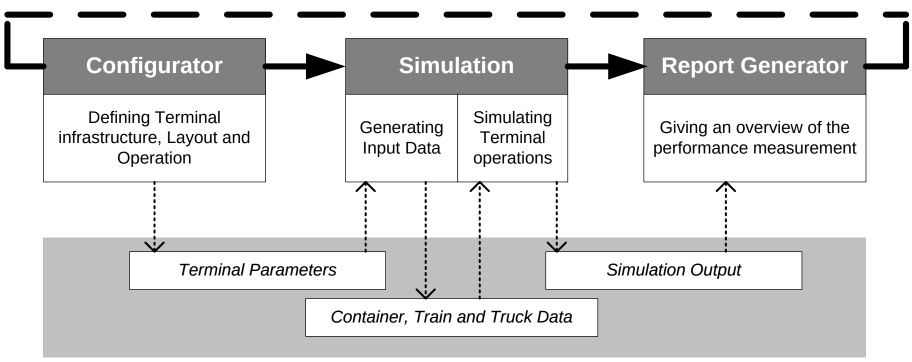


Figure 2. SimConT modules and data generation

Several authors propose analytical approaches, mainly MILP models for selecting various design options for rail-rail and rail-road terminals. Boysen et al. (2010) propose a dynamic programming approach, to determine yard areas for gantry cranes for balancing workload in order to improve the operation of the cranes. Wiese et al. (2011) describe different technologies in container terminal operation and their impact on the terminal layout. For a layout which is typical for the use of automated rail-mounted gantry cranes they propose a procedure to calculate promising storage yard configurations. Simulation as an evaluation method has also been studied intensively. Studies can be grouped into two categories. The first category concentrates on a certain subarea (see Yang et al. 2004), while the second category models the whole container terminal (see Gambardella et al. 1966; Lee et al. 2003; Parola and Sciomachen 2004 and Ballis and Golias 2004). This last category is rather comparable with our approach (Benna and Gronalt 2008). But due to the fact that nearly all relevant papers are devoted to open-sea Terminals, activities around the ship berthing, loading and unloading play a predominant role in the studies and are reflected in the process of goal setting, which is less suitable for our purpose.

In fact, in CTs activities and goals are rather centered on container shipment by railway. To summarize our literature review, it should be mentioned that only few research is done on the special nature of inland container terminals.

2 SIMCONT – ELEMENTS

We have both developed a simulation system (SimConT) and a procedure how to stepwise evaluate the capacity of a particular inland container terminal. The approaches are interdependent and we will now first present the SimConT elements and further show how we are using them on a detailed simulation case study. SimConT is both a concept and a simulation tool completely coded with Java Classes and supplemented by AnyLogic statecharts functions.

2.1 Terminal Configuration

The terminal configuration prepares the detailed layout of the terminal, the gates, parking areas inside and outside the terminal, loading and storage areas and all the distance and time related data. This is completed by opening and operating hours of the terminal. For the terminal equipment it is necessary to define its operations modes and ranges. For this an assignment of tracks and storage areas has to be defined and detailed operation plan for each of the various equipments is applied.

The terminal is dominated by the train schedule and the flow attributes of the ITEs. The daily arrival distribution of trains, their lengths and ITE-mix must be defined clearly and flexible for further adjustments. Also the operation mode (floating or fixed) for the trains generally and/or in particular effects the terminal capacity and must be provided as a system's input. Fig.3 shows the activity diagram of exporting containers with trucks and trains.

For particular ITEs the in- and outflow in the system, planned storage time, dedicated storage areas, stacking attributes and required handling equipment must be defined. The above presented data and dependencies are stored and provided in a database outside the simulation. It used as a generic input for the simulation and is application driven adapted. For example, it is necessary to work with a variable portion of non stackable trailers in inland CT due to an increase of intra-european freight traffic.

2.2 Simulation Model

The simulation model for conducting various experiments is coded with Any Logic, where the main control processes are implemented in state chart logic. As depicted in the first phase of the simulation is generating input data for the simulation model. This iteration makes the simulation application specific. According to the terminal configuration database the main elements of the simulation logic can be divided into Flow control, Handover control, Track control, Equipment control and Storage control.

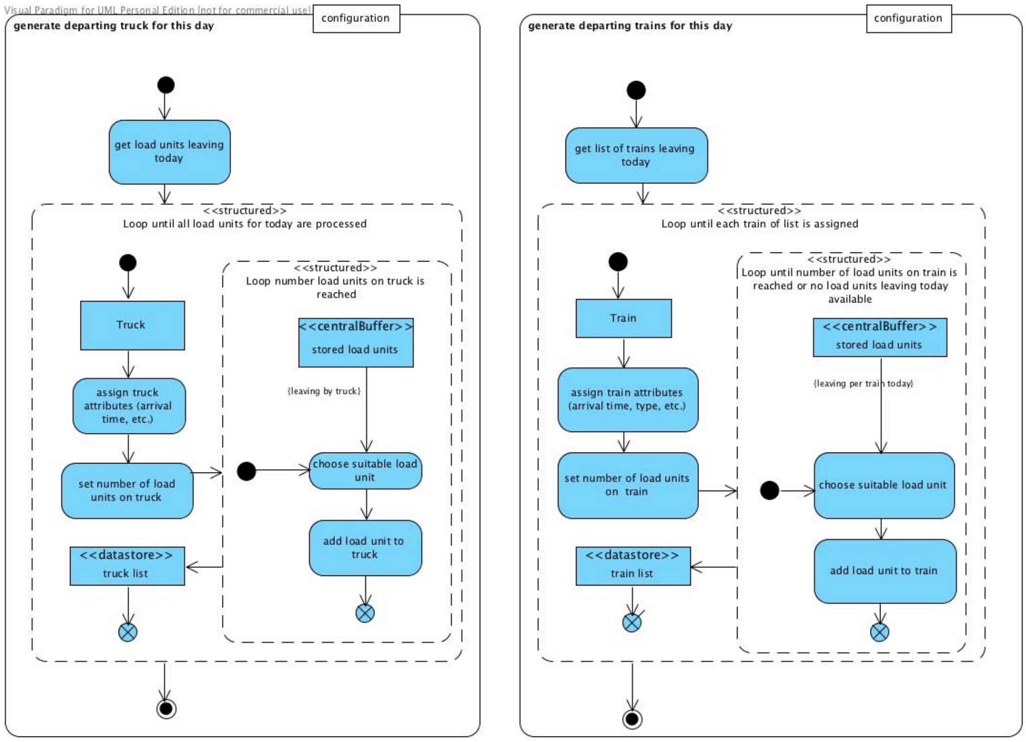


Figure 3. Activity diagram of generating trucks and trains

The left part of Fig.4 provides the storage control state chart. It selects the next possible storage spot for the specific ITE at hand. Clearly, the simulation model provides spots classes for controlling the storage areas.

In the right part of the figure we show the state chart for the equipment class. It displays the state transitions of cranes in the terminal. The supporting evaluation functions like pickTime(), dropTime(), chooseNext() which are responsible for the efficient movement of the cranes are not shown here. We have validated our approach during the conceptual phase with three different CTs. One further essential element is the simulation output generation which is presented in the next section.

2.3 Simulation Report

The simulation report puts together the simulation expertise with the CT process knowledge to extract the most relevant performance figures. For SimConT we decided on the following concept on report generation. Basic Indicators describe the CT in general. Regarding the terminal infrastructure we focus on the following data: storage capacity of loaded containers, empty container, trailers and swap bodies, total storage capacity, number and length of tracks, number of cranes per terminal unit, number of stackers, number of gates and number of container handover places. For the infrastructure related indicators usually utilization measures are evaluated in the reports. These can be defined both static and time dependent. Especially for the cranes the portion of service lifts are reported. In addition the terminal operation and container flow are also considered in the standard report. A basic set of indicators contain the number of import and export container per time unit and the portion of train and truck deliveries. According to these data, flow time indicators for containers, trucks and trains are reported. The set of basis indicators may be extended according to specific requirements of an application. For example for new terminals the truck queue in front of the gates are important to estimate the traffic jams caused by the terminal.

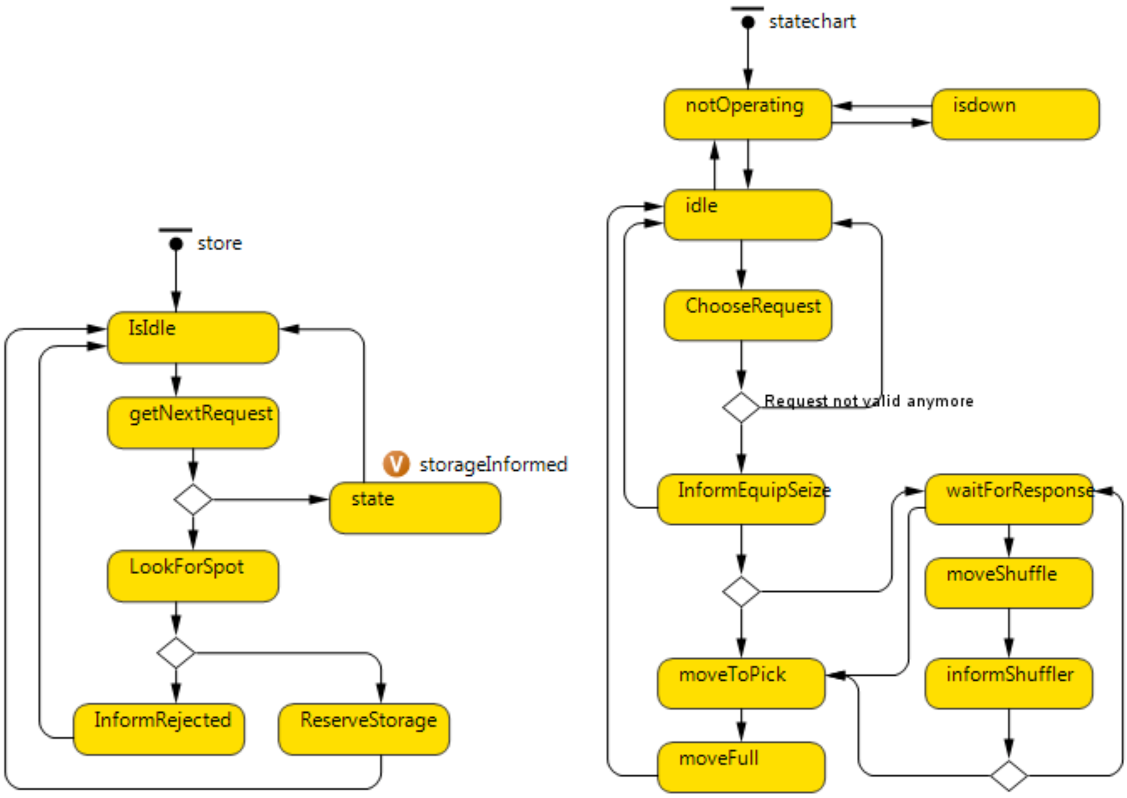


Figure 4. State charts (l) Storage allocation (r) crane – Simulation Logic

3 SIMULATION STUDIES

Simulation experiments are used to evaluate system's performance over time. Beside statistical issues like simulation run length, warm-up periods and number of replication and others it is further essential to consider specific terminal requirements. These may be related to the role of the terminal in the freight network (gateway, feeder, hinterland, industry supply) or the portion of empty container for exchange with industry or development possibilities of the terminal like additional terminal units. According to the role of the terminal we found different procedures on how to improve terminal's performance. In the first step of the simulation study we use the generic SimConT models as shown in Fig. 2 in order to simulate a Basis Scenario. Before we conduct the simulation experiments we apply the configuration steps in order to generate the data for terminal infrastructure, layout and operations strategies. For these it is especially important to build eligible train schedules. The data generated were validated with terminal operator and it also assists in defining the goals and varying parameters to guide the simulation experiments. For these the standard reports are extended to fit the simulation application. Usually,we use order data for one month as model input and generate the simulation data for the simulation period (e.g. one year).

The second step usually consists of the first simulation experiments for a new or existing terminal to calibrate the simulation to application specific restrictions. The simulation results (report) are analysed for critical or near critical performance values. If the model is calibrated we can now define future simulation scenarios. Some elements which may be used to define scenarios are listed below: train length, arrival frequency of trains, relation of direct exchange containers, dedicated storage for containers/swap bodies/trailers, number of terminal tractors and portion of liftable trailers. This iteration - definition of scenario parameters and simulation - is run again and again until the performance shows stable results and no further infrastructure options should be considered.

4 CAPACITY ANALYSIS OF INLAND CONTAINER TERMINAL – APPLICATION CASE

The starting point for this is the renewal of the Inland CT network in Austria (see Gronalt et al. 2010). For this, a performance analysis of existing and new terminals is required. We discuss the procedure to support the infrastructure investment decisions for a specific industry supply terminal, where the terminal handles a large amount of empty containers. The current configuration of the terminal was used as a baseline and compared to two improved terminal layouts. By doing this we defined some scenarios with increased transshipment volumes in order to figure out the marginal capacity of the terminal. The key figures of the starting configuration and the simulated layout variants are summarized in Tab. 1. One can see that Layout 2 provides considerably increased capacity in terms of storage areas and lifting resources. All terminal layouts were calculated with a throughput volume of 50000 ITE and we could observe that neither the current layout nor layout 1 were able to handle the volume.

Table 1. Basic terminal Figures

	Current Layout	Layout 1	Layout 2
storage capacity loaded container (TEU)	570	479	1277
storage capacity empty container (TEU)	1450	2773	3582
total storage capacity (TEU)	2020	3252	4858
number of tracks	6	4	4
track length	1310	2000	2240
cranes	1	2	2
stacker	2	2	2
truck parking area	9	15	20
gates	1	2	2

Fig. 5 displays the layout and storage blocks arrangement for terminal layout 2. Handover spots are located near the tracks for direct rail-truck transshipment. The number of transshipment points is restricted and also the number of trucks inside the terminals is controlled in order to prevent jams in the space of the terminal. Empty containers are stored in the area below the rail tracks and are operated by reach stackers. For terminal layout 2 an increase in transshipment volume is simulated in order to determine the marginal capacity of this layout. The volume increase is modelled both by a higher train arrival rate and by the number of containers per train.

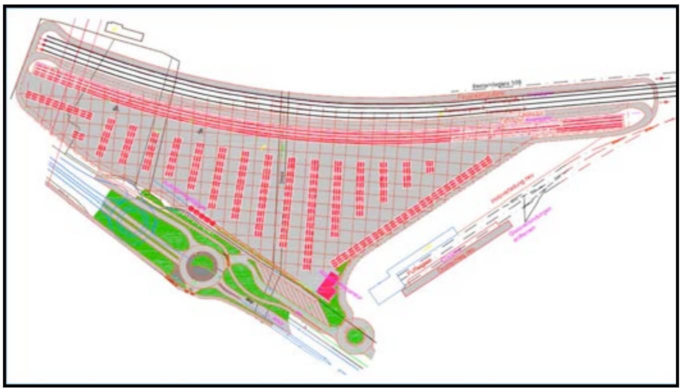


Figure 5. Terminal Layout 2

We develop a scenario tree in order to guide our analysis. Fig. 6 shows all 12 evaluated scenarios. Scenario A reflects the baseline. The B scenarios B1-B3 consider different order arrival and volume increase patterns for the given layout and operation strategy. The stepwise volume increase is due to the discrete nature of inserting new trains or increasing the number of containers per train. For all scenarios we keep the load carrier mix constant. The portion of non stackable containers accounts 17%.

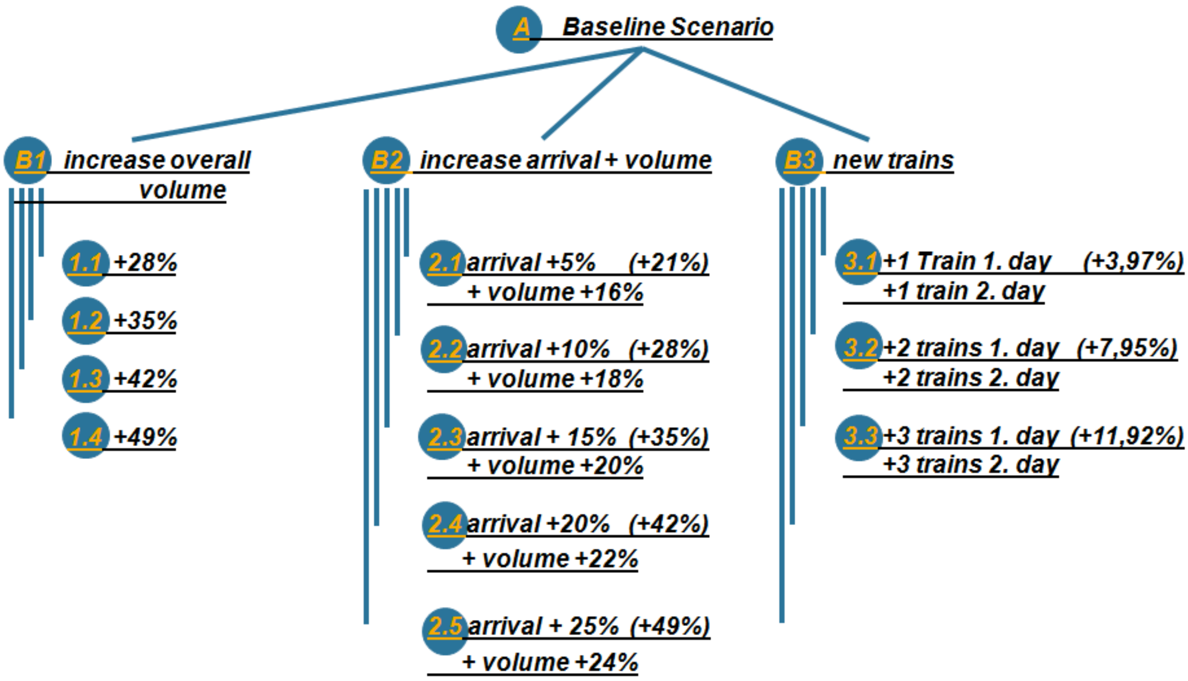


Figure 6. Scenario tree

For each scenario a simulation was made with a run period of one year for smoothing seasonal fluctuations in the results. The simulation runs were replicated 20 times to eliminate stochastic disturbances. We have developed a dedicated set of indicator values (resource utilization, storage usage, track usage, handover usage and waiting times for trains and trucks) for comparing the results. Tab. 2 shows some indicators of our simulation results. It can be noticed that an increase of 49% in volume will lead to very high crane utilization. On the other hand storage allocation is not critical. The available handover places are limiting terminal's capacity if volume is increasing to 35% or more.

Table 2. Comparison of Scenario B1 with the basis Scenario A

	Scenario				
	A	B1			
	50.000 ITE	+28%	+35%	+42%	+49%
full container storages	0,29	0,29	0,30	0,34	0,33
empty container storage	0,28	0,29	0,31	0,30	0,39
crane utilization	0,40	0,51	0,56	0,63	0,76
track utilization	0,57	0,62	0,70	0,75	0,78
handover places used at same time	14,00	16,00	18,00	18,00	18,00
truck parking utilzation	0,45	0,55	0,58	0,58	0,64

As a further example, Fig.7 shows the crane utilization rates for different volume increase scenarios B2. It can be seen that for the current volume which is at the limit of the existing terminal's capacity, the new layout will be able to handle these volume very easily. But a 42%

increase or above leads to near critical crane average utilization of about 70%. We also can notice that we may have some opportunities to raise the capacity if we can change train schedules.

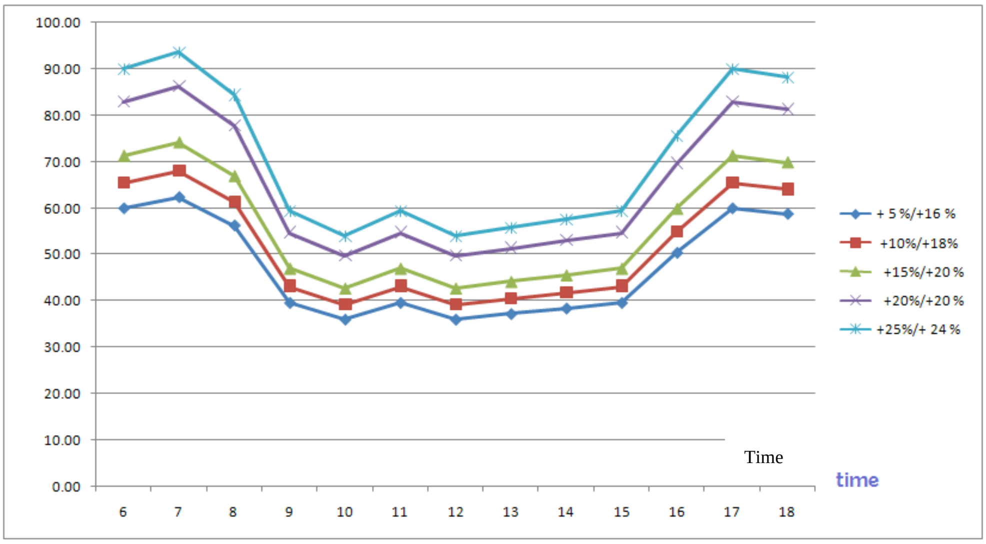


Figure 7. Daily development of crane utilization for B2 scenarios

In scenario B3 a discrete insertion of new trains is modelled in addition to the basis scenario A. Days with minimum workload were selected for the insertion of the new trains. The results in Tab. 3 show that these new trains can be handled by the terminal.

5 CONCLUSIONS AND FUTURE RESEARCH OPTIONS

In this paper we presented our point of view in conducting simulation studies with the generic inland CT simulation tool SimConT. With the developed modelling approach and simulation tool we are able to evaluate the performance of inland container terminals. Especially during design of new sites it is of importance to support decision makers with both quick and accurate results of particular terminals layout and future loading carrier mix scenarios. Actual and future work is on transferring the modelling environment to a more complicated environment of multi-unit terminals.

Table 3. Scenario B3

	Scenario			
	A	B3		
		8 trains	16 trains	24 trains
Volume	50.000 ITE	+3,97%	+7,95%	+11,92%
full container storages	0,29	0,29	0,29	0,29
empty container storage	0,28	0,29	0,29	0,29
crane utilization	0,40	0,43	0,47	0,50
track utilization	0,57	0,58	0,59	0,61
handover places used at same time	14,00	14,00	18,00	18,00
truck parking utilzation	0,45	0,45	0,46	0,48

REFERENCES

Ballis, A. and Golias, J. (2004), Towards the improvement of a combined transport chain performance. *EJOR* **152 (2)**, 420-436.

Benna, T. And Gronalt, M. (2008), Generic Simulation for Rail-road container terminals. *Proceedings of the 2008 Winter Simulation Conference*, pp. 2656 – 2660.

Boysen, N., Fliedner, M. and Kellner, M. (2010), Determining fixed crane areas in rail-rail transshipment yards. *Transportation Research Part E: Logistics and Transportation Review* **46(6)**, 1005-1016.

Gambardella, L. M., G. Bontempi, E. Taillard, D. Romanengo, G. Raso, and P. Piermari. (1996), Simulation and Forecasting in Intermodal Container Terminal. Available via http://www.idsia.ch/~luca/ess96.pdf> [accessed Mai 4, 2008].

Gronalt M., Höfler L., Humpl D., Käfer A., Peherstorfer H., Posset M., Pripfl H., Starkl F.: *Handbuch Intermodaler Verkehr*. Bohmann Verlag, Wien 2010

Lee T.-W., Park N.-K., and Lee D.-W. (2003), A Simulation study for the logistics planning of a container terminal in view of SCM. *Maritime Policy & Management* **30 (3)**, 243-254.

Parola, F., and Sciomachen, A.(2004), Intermodal container flows in a port system network: Analysis of possible growths via simulation models. *International Journal of Production Economics*, 2-14.

Franc, P. and Van der Horst, M. (2010), Understanding hinterland service integration by shipping lines and terminal operators: a theoretical and empirical analysis. *Journal of Transport Geography*, **18(4)**, 557-566.

Wiese, J., Suhl, L. and Kliewer, N. (2011), Planning Container Terminal Layouts Considering Equipment Types and Storage Block Design. In *Handbook of Terminal Planning*, 49:219-245. Operations Research/Computer Science Interfaces Series. Springer New York, 2011.

Yang, C. H., Y. S. Choi, and T. Y. Ha. (2004). Simulation based performance evaluation of transport vehicles at automated container terminals. *OR Spectrum* ***26***, 149-170.

MULTI-OBJECTIVE CONTAINER STORAGE ARRANGEMENT FOR TRANSSHIPMENTS

Etsuko Nishimura[1] and Yoshiyasu Urushibara[1]

[1]*Graduate School of Maritime Sciences, Kobe University, 5-1-1 Fukae-minami Higashinada, Kobe, Japan, e-nisi@maritime.kobe-u.ac.jp*

ABSTRACT

This study addresses the container storage arrangement on the yard with various number of block partitions. We consider the minimization of both total handling time and yard space used as objective functions. This problem is formulated as a mixed integer programming, we re-arrange the basic multiple objective tabu search (MOTS) which Hansen(1997) proposed in order to obtain the solutions. From comparing the results with those by single objective with minimizing handling time, in all cases, the time and space utilizations obtained by multi-objective are smaller than those by single one. For the continuous berthing pattern, the greater the number of partitions is, the shorter the time becomes. However, for the discrete berthing pattern, we obtain the shortest time in the order 16 and 8 partitions at four-berth terminal. This means that all berthing positions for the discrete pattern are located at close to the paths among yard blocks with 8 and 16 partitions, and the yard trailers do not have to go a long way for keeping one way. It is clear that the relationship between berthing location pattern and number of partitions effects on the yard trailers' routing, and more that is also do that on the total handling time and block utilization.

Keywords: container storage, multi-objective, tabu search

1 INTRODUCTION

There has been shown to be a tendency towards great numbers of the container throughput as the world trade expands. The mega-containerships with the capacity of over 10,000 TEU are serviced between Asia and Europe. There are a hundred and over mega-containerships at the middle of 2011. At a terminal where mega-containerships are calling, most cargoes to be handled transfer from origin to destination ports via that terminal. Thus, associated handling operations are undertaken between mega-containerships and feeder ships. In order to decrease the turnaround time of ships calling at this terminal, smooth handling of operations is of major importance.

Most of container terminals in major ports of Japan are constructed on the land reclaimed from the sea in order to be satisfied with the constraints between water depth and ships' draft. Those terminals have a rectangle shape composed from multiple-berth into a line. However, there are also the terminals which have irregular shapes of quayside and landside because of geographical factors in the world. At the terminal with an irregular shape, the positions and total number of paths located among container blocks have influences on the yard trailers' movement. Furthermore, yard trailers' behaviour has an influence on the time spent for containers handled. At this step in our study, for the terminal with a simple of both shape and layout, we are concerned with the container storage planning problem at the terminal with various number of yard block partitions, in order to carry out the container handling operations in the most efficient way.

This paper is organized as follows. The next section reviews the related literature. In Section 3 the problem is defined. In sections 4 and 5, the problem formulation and the solution procedure are described, respectively. In section 6 numerical experiments are carried out and computational results are presented. The final section reports the paper’s findings and conclusions.

2 LITERATURE REVIEWS

Storage space is a critical resource in container terminals. Kozan and Preston (1999) consider the determination of optimal storage strategies and container handling schedules by yard cranes. They propose a heuristic method by a genetic algorithm and analyse the factors that influence container transfer efficiency through lower throughput time at a container terminal with different types of handling equipment, storage capacities and layouts.

Kim *et al.* (2000) propose a solution method for determining the storage location of export containers considering their weights. They consider the stack arrangement and the weight distribution of containers in a yard bay. They develop a dynamic programming model for determining the storage allocation, in order to minimize the number of relocation movements. Kim and Park (2003) discuss how to allocate the storage space for outbound containers. They formulate the model as a mixed-integer linear program in order to allocate the space for utilizing spaces efficiently and make loading operations more efficient. Two heuristic algorithms are proposed.

Kim and Hong (2006) address the problem of relocating containers in the yard block. They propose a branch and bound algorithm and a heuristic rule based on probability theory. And they also propose a procedure for estimating the expected additional number of relocations which can apply various configurations of stacks. Yang and Kim (2006) address the storage space to containers assignment problem so that the total number of relocations expected is minimized. Container treatment is grouped as storage demand unit by destination port, weight and size classes. They consider the static problem in which arrival and departure times of containers are known, the dynamic problem in which arrival time of containers are not known. They solve the static problem using a dynamic programming technique and a genetic algorithm, the dynamic problem is solved by their proposed heuristic methods.

Zhang *et al.* (2003) address the storage space allocation problem in the yard. They solve the problem using a rolling-horizon approach. This problem is decomposed into two levels: placing the total number of containers in each block at the time period, and determining the number of containers associated with each vessel in order to minimize the total distance to transport the containers between yard blocks and the ship berthing location. They deal with four types of containers as follows: inbound containers on vessels and already stored in yard blocks, outbound containers grounding in yard blocks and already stored in yard blocks. However Bazzai *et al.* (2009) consider only inbound containers at the first level of the problem that Zhang *et al.* (2003) have dealt with, they determine the number of inbound containers of each vessel stored in each block, in order to minimize the imbalance among workloads allocated to blocks.

Lee *et al.* (2006) consider the problem in order to determine the minimum number of yard cranes to deploy and the location where the group consisted by unloaded containers should be stored. They have formulated the problem as a mixed integer-programming model, in order to minimize the total number of shifts required to handle all the workload. They also have proposed two heuristics: a sequential method and a column generation method. Han *et al.* (2008) consider the container-to-yard location assignment problem, as given the concept to reserve dedicated clusters for each vessel. They determine the number of containers discharged during each period, in order to minimize the total number of yard cranes required to handle the workloads. Lim *et al.* (2006) propose a meta-heuristic procedure, named the critical-shaking neighbourhood search, in order to minimize the yard space needed, as given for yard spaces to requests at each time.

Park and Seo (2009) consider the planar storage location assignment problem which can be defined as the assignment of the inbound and outbound containers to the storage yard with aim of minimizing the number of obstructive container moves. They proposed an efficient genetic algorithm to solve the problem for real-sized instances. Park and Seo (2010) also consider the problem as well as Park and Seo (2009). They proposed a genetic algorithm based heuristic and their original heuristic methods. In order to consider three conditions that make blocks move as following: (a) retrieving due outbound blocks from the yard, (b) replacing obstructive blocks in the path of retrieving the due outbound block to other empty space, and (c) storing inbound blocks at the yard, their original heuristic is consisted by two independent sub-algorithms: (1) obstructive block selection and (2) storage location determination.

Recently, there are the studies dealt with containers at transshipment hubs as Lee *et al*. (2006) and Han *et al.* (2008). They do not consider the container flow at the terminal between the mega-containership and feeder ships. Then Nishimura *et al*. (2009) consider the problem for transshipment containers at the terminal where the mega-containership calls. However, this study deals with minimizing the service time with the container flow between only one mega-ship and multiple feeders. Therefore, in order to generalize the problem, this study multiple objective functions including space utilization with the container flow among multiple ships.

3 PROBLEM DEFINITIONS

The terminal where the mega-containership calls is not only used by that ship and feeders but also done by ships with other sizes. The yard area where the containers stored is divided roughly according to the direction of containers passing through that terminal such as inbound, outbound and transshipment. This paper deals with the problem for only transshipment containers related to all ships.

Secondly, we describe the treatment of containers handled. A yard area has multiple container blocks, and one container block consists of multiple containers. In this study, containers with weight class and destination port are treated as a particular container group. For instance, a container group for ship “A”, which consists of one hundred containers, is stored in any container block in the yard. Each container group is given the inbound and outbound ships in advance. After the relevant container arrives from inbound ship, it stays in the terminal until outbound ship arrival. And then, the relevant container group is loaded to outbound ship,

Thirdly, we state how to use a single block. Generally, the multiple containers handled can share a specific block if the time of one ship serviced is different from that of other ships serviced. However Nishimura et al. (2009) assume that a part of the yard area in the target block is used for the containers handled of other ships in another planning period, in order to simplify the model. Therefore, in order to generalize the model, we assume that multiple container group can share a specific block simultaneously.

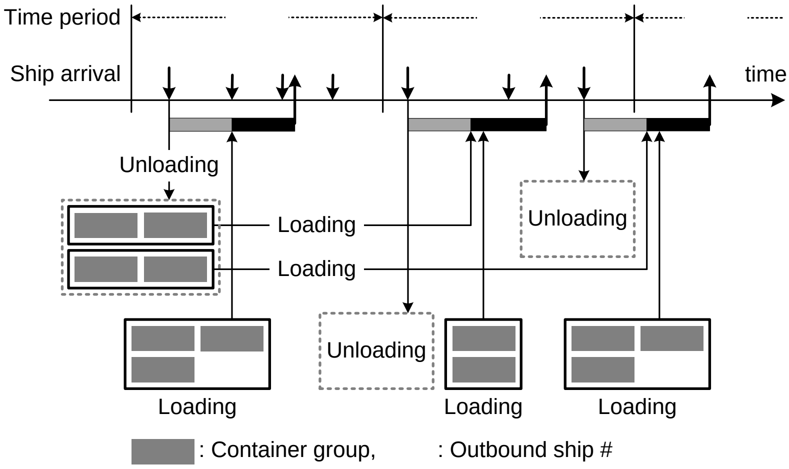


Figure 1. Containers flow on chronological order

Fig.1 shows the container flow on chronological order. Horizontal arrow means the time index. " t " means a planning time period. We determine the storage location for containers unloaded from the ship arriving during the relevant period. As the initial situation, we assume that some containers have already been stored in the yard area. Those containers are planning to be loaded to the ships which arrive from now on. Ships do not always arrive simultaneously, but we do not consider the difference between arrival times of those ships in a specific period. For instance, ships 1 to 4 arrive within the period t=1, and ships 5 to 7 do within t=2. The storage locations for containers unloaded and loaded to those ships are determined in advance. The times spent for unloading and loading depend on the storage location of the relevant container, the departure time for its related ship is also determined. In this example, ships 1 and 5 leave during the same period as the time of those ships arrival. However ship 7 leaves during next period to the time of this ship arrival. Containers unloaded from ship 1 are for containers loaded to ships 5 and 7. Those containers are loaded to ships 5 and 7 with other containers which have already been stored in the yard area.

In this study, we will conduct the numerical experiments under various number of yard block partitions. This also changes the storage capacity. Thus if we can increase the ratio of block's share, the containers except transhipment can use the other blocks without target containers' use. From this reason, we consider the objective functions: (a) minimization of handling time and (b) minimization of space utilization.

The determinant, the objective function and constraints of our problem are as follows:

the determinant is a storage allocation for transshipment containers in the yard,

the objective function is total handling times for target ships,

constraints are

– each container group is assigned to any yard block exactly once, and
– multiple container group can be stored at a yard block simultaneously if total number of those does not exceed the capacity of yard block which containers are assigned.

4 PROBREM FORMULATIONS

In this section, we show a mixed integer programming formulation of the container storage allocation problem (CSAP) as follows:

Parameters

$t(=1,\dots,NT)\in T$ set of planning periods

$k(=1,\dots,NG_t)\in CG_t$ set of container groups, which is handled during time *t*

$g(=1,\dots,NG(=\sum_{t\in T}NG_t))\in CG(=\sum_{t\in T}CG_t)$

set of container groups, which is handled whole planning period

$l(= 1, \ldots, NL) \in YB$ set of yard blocks

J_g set of inbound ship for container group g

J_g^- set of outbound ship for container group g

$B(J)$ berthing position of ship J

S_{J_g} beginning time for containers handled to ship J_g

A_{tJ_g} 1 if ship J_g arrives during time t, 0 otherwise (0-1 integer values)

$C_{B(J_g)l}$ handling time spent from berthing position of ship J_g to yard block l

CAP_l capacity of yard block l

v_{qg} 1 if quay crane q is assigned to container group g, 0 otherwise (0-1 integer values)

F_t end of planning horizon t

M very large value

Variables

x_{gl} 1 if container group g is stored at yard block l, 0 otherwise (0-1 integer decision variables)

y_{tl} number of container groups, which are occupied in yard block l at beginning time of planning period

d_{J_g} departure time of ship J_g

$(\alpha_{tJ_g}, \beta_{tJ_g})$ (1,0) if ship J_g departs from the relevant terminal during time t, (0,0), (0,1) or (1,1) otherwise (0-1 integer decision variables)

Formulations

$$\text{Minimize} \quad \sum_{g \in CG} \sum_{l \in YB} \{ C_{B(J_g)l} + C_{B(J_g^-)l} \} x_{gl} \tag{4.1}$$

$$\text{Minimize} \quad \max_{t \in T} \sum_{l \in YB} (y_{tl} + \sum_{g \in CG} A_{tJ_g} x_{gl}) / NL \tag{4.2}$$

Subject to

$$\sum_{l \in YB} x_{gl} = 1 \qquad \forall g \in CG \tag{4.3}$$

$$y_{tl} + \sum_{g \in CG} A_{tJ_g} x_{gl} \le CAP_l \qquad \forall t \in T, l \in YB \tag{4.4}$$

$$d_{J_g} = S_{J_g} + \max_{q \in QC} \{ \sum_{g \in CG} \sum_{l \in YB} C_{B(J_g)l} x_{gl} v_{qg} \} \qquad \forall J_g \in V \tag{4.5}$$

$$d_{J_g} < F_{t-1} + M\alpha_{tJ_g} \qquad \forall t \in T,\ J_g \in V \tag{4.6}$$

$$F_{t-1} < d_{J_g} + M(1 - \alpha_{tJ_g}) \qquad \forall t \in T,\ J_g \in V \tag{4.7}$$

$$d_{J_g} \le F_t + M\beta_{tJ_g} \qquad \forall t \in T,\ J_g \in V \tag{4.8}$$

$$F_t \le d_{J_g} + M(1 - \beta_{tJ_g}) \qquad \forall t \in T,\ J_g \in V \tag{4.9}$$

$$\alpha_{tJ_g} + \beta_{tJ_g} = 1 \qquad \forall t \in T,\ J_g \in V \tag{4.10}$$

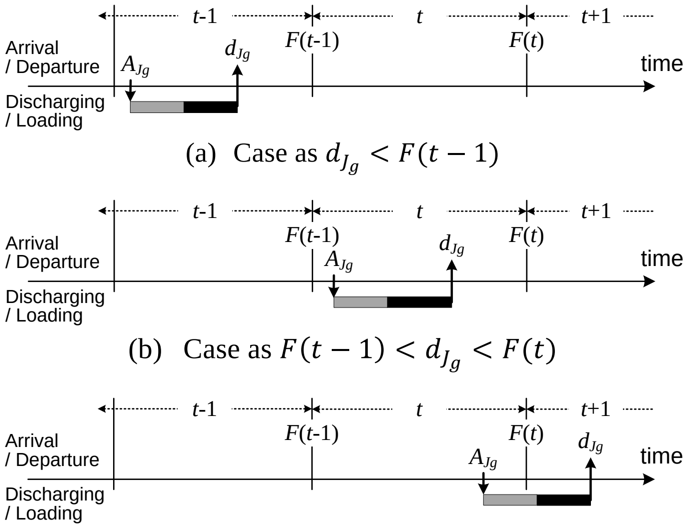
$$y_{t+1,l} = y_{tl} + \sum_{k \in CG_t} A_{J_k} x_{kl} - \sum_{g \in CG} \alpha_{tJ_g} x_{gl} \qquad \forall t \in T,\ J_g \in V,\ l \in YB \tag{4.11}$$

$$x_{gl} \in \{0, 1\} \qquad \forall g \in CG, l \in YB \tag{4.12}$$

$$y_{tl} \ge 0 \qquad \forall t \in T, l \in YB \tag{4.13}$$

$$d_{J_g} \ge 0 \qquad \forall J_g \in V \tag{4.14}$$

$$\alpha_{tJ_g},\ \beta_{tJ_g} \in \{0, 1\} \qquad \forall t \in T, J_g \in V \tag{4.15}$$



(a) Case as $d_{J_g} < F(t-1)$

(b) Case as $F(t-1) < d_{J_g} < F(t)$

(c) Case as $F(t) < d_{J_g}$

Figure 2. Relationship between variables $F(t)$ and d_{Jq}

Objective function (4.1) shows minimizing the sum of handling times spent between ship berthing location and container stacking area, and (4.2) also shows that the minimization of the yard block utilization. Constraints (4.3) mean that each container group must be stored in any yard block exactly once. Constraints (4.4) ensure that some container groups can be stored in the relevant yard block, if the sum of those is not more than the capacity of yard block. Constraints (4.5) define the departure time of each ship. That time sets to the latest completion time which the quay crane is handled if multiple quay cranes are assigned to the relevant ship. Constraints (4.6) to (4.10) show that the relationship among variables d_{Jg} and $(\alpha_{tJ_g}, \ \beta_{tJ_g})$. Constraints (4.11) define that the initial situation of period (t+1) after the empty of yard space is updated at period t.

We explain constraints (4.6) to (4.10) in detail from Fig.2. In the case of (a), constraints (4.6) to (3.9) are satisfied if variables $(\alpha_{tJ_g}, \ \beta_{tJ_g}) = (0, 0)$. However, constraints (4.10) are not done that. In the case of (b), constraints (4.6) to (4.9) are also satisfied if variables $(\alpha_{tJ_g}, \ \beta_{tJ_g}) = (1, 0)$. Additionally, in the case of (c), constraints (4.6) to (4.9) are satisfied if variables $(\alpha_{tJ_g}, \ \beta_{tJ_g}) \ = \ (1, 1)$, (4.10) are not done that.

From above (a) to (c), constraints (4.6) to (4.10) can be satisfied simultaneously under only (b) which the relevant ship departs from the terminal during the relevant time t.

5 SOLUTION PROCEDURE

As shown in section 4, CSAP can be formulated as a mixed integer programming. In order to solve this problem, we propose the heuristic procedure by tabu search (TS).

5.1 Algorithm by using TS

As stated in Glover and Laguna (1997), the emphasis on responsive exploration in TS, whether in a deterministic or probabilistic implementation, derives from the supposition that a bad strategic choice can yield more information that a good random choice. Responsive exploration integrates the basic principles of intelligent search. TS is concerned with finding new and more effective ways of taking advantage of the mechanisms associated with both adaptive memory and responsive exploration.

The fundamental mechanism of TS is that potentially reversing the effects of previous moves by interchanges that might return to previous positions, in order to prevent the search from repeating swap combinations tried in the recent past. The memory structures in TS operate by reference to following dimension consisting of recency, frequency, quality and influence. The most commonly used short term memory keeps track of solutions attributes that have changed during the recent past, and is called recency-based memory or tabu list. To exploit this memory, selected attributes that occur in solutions recently visited are labeled tabu-active, and solutions that contain tabu-active elements, or particular combinations of

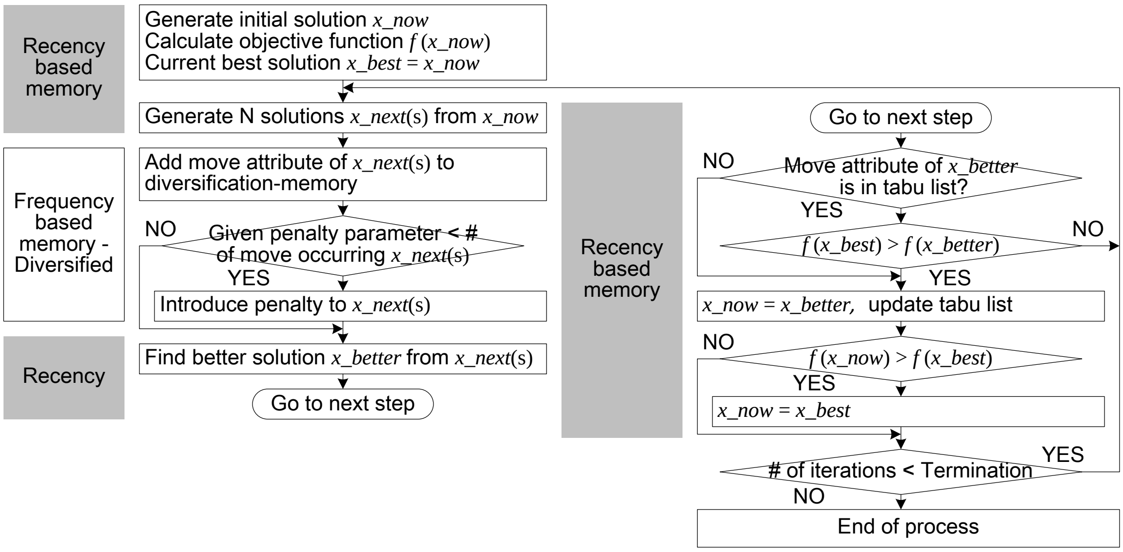


Figure 3. Flow of TS

these attributes, are those that become tabu. Frequency-based memory provides a type of information that complements the information provided by recency-based memory, broadening the foundation for selecting preferred moves. Quality memory refers to the ability to differentiate the merit of solutions visited during the search. TS concept of quality is broader than the one implicitly used by standard optimization methods. Influence memory considers the impact of choices made during the search, not only on quality but also on structure. Recording information about the influence of choices on particular solution elements incorporates an additional level of learning. The flow of TS is shown in Fig.3.

5.2 Solution representation and move attribute

Fig.4 shows an example of the solution representation for two planning periods. Container groups 1 to 10 arrive during 1st period, and also 11 to 20 arrive during 2nd period. We determine the yard block randomly while finding empty spaces, in each period on step by step. In this example, container groups 1 to 10 are assigned to yard blocks 6, 8, 2, 7, 1, 12, 1, 3, 6 and 11, respectively. And also container groups 11 to 20 are done to 1, 8, 2, 7, 5, 5, 1, 3, 6 and 10, respectively. Each container group has the inbound and outbound ships which are given in advance. As shown in this example, containers 1 to 10 unloaded from inbound ships 1 to 4 in the relevant period are loaded to ship 5 and after coming ships. In fact, the representation consists of more time periods. Note that, from our knowledge, the period when containers are loaded to the outbound ship is the next period, and after that, those are unloaded to the inbound ship.

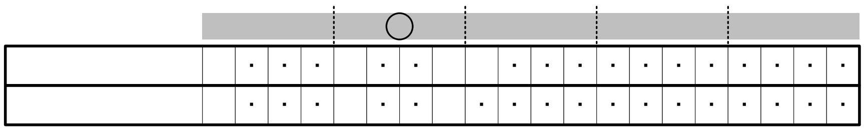
In order to generate the new solution, as shown in Fig.5, we can arrange the past solutions by swapping movement. Firstly, we select just one planning period randomly. Next, we select two container groups, and swap move them. If new solution cannot be satisfied with the constraint about block capacity, we find the yard block of container group randomly until being satisfied with that constraint. The number of candidate solutions generated from the current solution at each iteration is set to 100, and the termination is set to 500.

5.3 Recency and frequency based memories

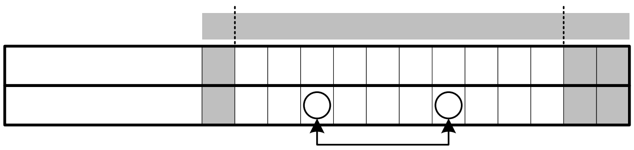
The recency based memory respectively identify the starting and ending iterations of the tabu tenure for specified attribute “ *e* ”, thus bracketing the period during which “ *e* ” is tabu-active. This tabu tenure is called as tabu list size in most of related papers. In this paper, tabu list size is set to 10.

Inbound ship #	1	1	1	2	2	2	3	3	4	4	5	5	6	6	6	7	7	8	8	8
Outbound ship #	5	7	7	5	5	6	7	7	5	10	11	9	12	10	12	13	9	11	9	11

Figure 4. Solution representation



(a) Select the planning period randomly



(b) Swap move in the relevant planning period

Figure 5. How to obtain new solution while keeping constraints

Frequency based memory provides a type of information that components the information provided by recency based memory, broadening the function for selecting preferred moves. The role of intensification and diversification in tabu search become especially relevant in longer term search processes. Intensification strategies undertake to generate solutions by aggressively encouraging the corporation of good attributes. In intermediate to long term, it consists of incorporating attributes of solutions from selected elite subsets. On the contrary, diversification strategies seek to generate solutions that embody compositions of attributes significantly different from those encountered previously during the search. These two types of strategies counterbalance and reinforce each other in several ways.

However, as consequences of some test cases in this model, intensification strategies do not have effects on the results. Therefore, we apply only diversification as the following equations proposed in Glover and Laguna (1997b) are used in this paper.

$$\varepsilon(H, S^*) = \begin{cases} f(S^*) & (\text{if } f(S^*) \le f(S)) \\ f(S^*) + \alpha \times Freq(i, j) & (\text{if } f(S^*) > f(S)) \end{cases} \tag{5.1}$$

where

H	recording information used in search process
S^*	new solution
$\varepsilon(H, S^*)$	function of frequency measurement for solution S^* considering with recording information H
$f(S^*)$	objective function value of new solution S^*
$Freq(i, j)$	frequency of the swap move (i, j) used until current iteration, in order to generate solution S^* by arranging solution S

α a fixed parameter for keeping the balance between objective function value and frequency used, this depends on range of objective function value, iterations, size of search history etc.

5.4 Multiple objective TS

In order to solve the optimization problem with two objective functions, in this study, we apply the multiple objective TS (Basic MOTS) proposed by Hansen (1997). MOTS algorithm re-arranged for this problem is as follows.

The symbols used in the following algorithm are as follows:

$x_i (i = 1, \dots, m) \in X$ set of candidate solutions
TL_i tabu list
ND set of non dominated solutions
N_{xi} set of non-dominated solutions for solution x_i
$k \in K$ type of objective function (k=1 if minimizing of total handling time; k=2 if minimizing of block utilization(%); $|K|$ means the number of objective functions)
π^k range equalization factor of objective function k
λ_i^k weight vector for objective k of solution x_i
$f^k(x_i)$ value for objective function k of solution x_i
w_i^k proximity function obtained by the Manhattan distance norm between solution x_i and others (the shorter the distance is, the greater the value becomes.)

Step 1 generate solution $x_i \in X$ randomly
Step 2 initialize parameters: $TL_i = \emptyset$, $ND = \emptyset$, π^k=1/$|K|$ ($\forall\ k \in K$), iteration=1
Step 3 give solution number i=1
Step 4 weight vector λ_i^k=0 ($\forall\ k \in K$)
Step 5 find non dominated solution x_j for solution x_i , and set to $x_j \in N_{xi}$
Step 6 obtain proximity function w_i^k ($\forall\ k \in K$)
Step 7 update weight vector λ_i^k ($\forall\ k \in K$)
Step 8 generate new solution y_i to the solution which maximizes $\sum_{k \in K} \lambda_i^k f^k(x_i)$
Step 9 update TL_i as recording the attribute of solution y_i . Set to x_i=y_i .
Step 10 record solution y_i to ND and update the range equalization factor π^k,
Step 11 replace one solution and another selected randomly from X at regular intervals of iterations
Step 12 i=i+1, go to Step 4
Step 13 stop if iteration > termination. iteration= iteration+1 and go to Step 3 otherwise

In step1, initial candidate solutions are generated randomly. In step2, tabu list and set of non-dominated solutions are empty, and parameters like the range equalization and the counter are initialized. From steps 3 to 12, for each solution x_i, find the non dominated solutions and update parameters in order to find next solutions. From steps 4 to 7, the weight vector for solution x_i is determined. From Hansen (1997), this weight is set so that the solution moves away from the others, ideally having the solutions equidistantly spread over the frontier. The closeness is measured by a distance function based on some metric in the objective function space and using the range equalization weights. We can obtain as follows:

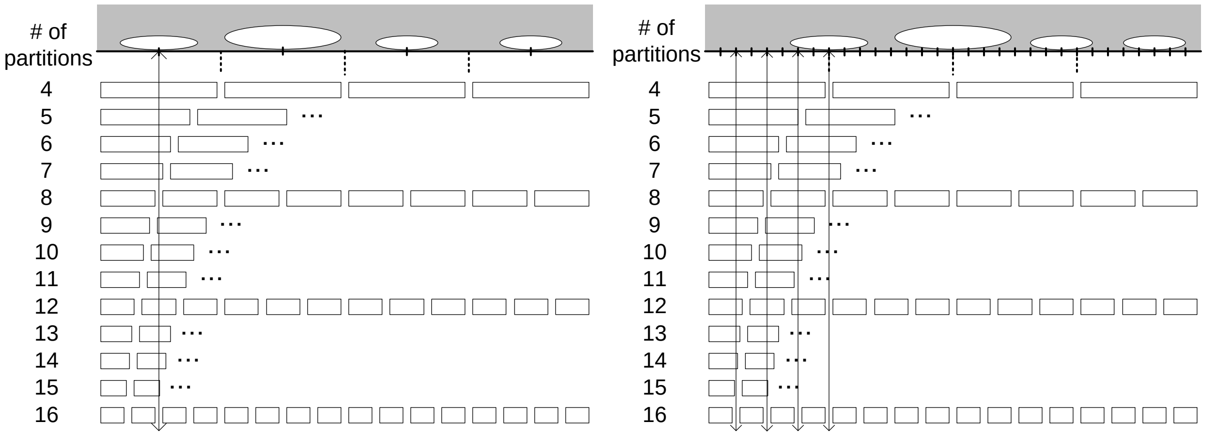
$$w_i^k = 1 / \sum_{j \in N_i} \pi^k |f^k(x_i) - f^k(x_j)| \quad \forall\ k \in K \tag{5.2}$$

$$\lambda_i^k = \lambda_i^k + \pi^k w_i^k \text{ if } f^1(x_j) < f^1(x_i) \text{ and } f^2(x_j) < f^2(x_i) \quad \forall\ k \in K \tag{5.3}$$

And we conduct the following operation, in order to normalize weight vector λ_i^k , set λ_i^k to 0-1 real random number if $\lambda_i^k = 0$ ($\forall\ k \in K$). Note that $\sum_{k \in K} \lambda_i^k = 1$.

From steps 8 and 9, the standard tabu search algorithm is operated. In step 10, new solution y_i can included in *ND* if it is non-dominated by all solutions in *ND*, and update the range equalization factor π_i^k, if $f^1(y_i) < f^1(x_l)$ and $f^2(y_i) < f^2(x_l)$ ($x_l \in ND$),

$$\pi_i^k = \frac{1/|f^k(x_i)-f^k(x_l)|}{1/|f^1(x_i)-f^1(x_l)| + 1/|f^2(x_i)-f^2(x_l)|} \qquad \forall k \in K\ ,\ x_l\ (\neq x_i) \in N \tag{5.4}$$



(a) Discrete location **(b) Continuous location**

Figure 6. Ship berthing positions and path's locations in each partition

In step11, replace one solution and another selected randomly from *X* at regular intervals of iterations, in order to ensure the movement of solutions over non-dominated frontier.

6 NUMERICAL EXPERIMENTS

6.1 Experimental design

The solution procedure is coded in the "C" language. Problems used in the experiments are generated randomly, but systematically. We consider the terminal layout with four berths consisted of a 400m berth. The types of yard block sections (we show it as the number of partitions) as our stated in Section 1, and the types of ship berthing location are shown in Fig. 6. Eight partitions are totally located in a block with conventional size, which are cut one berth length into half size. Thus on the basis of eight partitions, no division for one berth become totally four partitions. And more four divisions for one berth become totally 16 partitions. Then we consider the terminal with the yard block from 4 to 16 partitions. Additionally, yard trailers' routing depends on the distance between the ship berthing location and the paths located in yard area. Therefore, we consider two types as the ship berthing location: discrete and continuous versions shown in Fig. 6.

6.2 Computational results

Table 1 shows total handling time (hours) and yard block utilization (%) at each number of partitions, which has the computational results of discrete and continuous berthing locations, respectively. In order to validate the treatment of multi-objective including space utilization, we do not show only the results obtained by multi-objective (hereafter we call MOBJ), but also those done by single-objective (hereafter we call SOBJ) with only total handling time.

First of all, we compare the results obtained by SOBJ with those done by MOBJ. In all cases, the yard block utilizations by MOBJ are smaller than those by SOBJ. The less the number of partitions and the greater the one block size is, the greater the difference between block utilizations by SOBJ and MOBJ becomes. And as well, total handling times obtained by MOBJ are shorter than those done by SOBJ.

Secondly, we investigate the influences on results by number of partitions and berthing location patterns. Handling times decrease from 4 to 8 partitions in a monotone, there is not so significant among handling times obtained from 5 to 15 partitions, and handling time done by 16 partitions is shorter than the others. For the continuous berthing location, the greater the number of partitions, the shorter handling time become. However, for the discrete berthing location, the shortest handling is obtained at 16 partitions, and the second shortest handling time is done at 8 partitions. This means that all berthing positions in the discrete pattern are located at close to the paths among yard blocks of 8 and 16 partitions, and the yard trailers do not have to go a long way for keeping one way.

Thirdly, looking at the block utilization, there is not so significant the difference among the number of partitions. Actually, we obtain the smallest space utilizations on order at 10 and 8 partitions in discrete location, and also do those at 14 and 12 partitions in continuous location. The number of partitions defines the location of paths on perpendicular direction to quay length. Therefore, it is clear that the relationship between berthing location pattern and number of partitions effects on the total handling time and block utilization.

Table 1. Numerical results

(a) Berthing location pattern : ***Discrete***

# of partitions	Total service time (hours)					Yard block utilization (%)				
	SOBJ	MOBJ (Parete solutions)				SOBJ	MOBJ (Parete solutions)			
		Avg.	Std.	Min.	Max.		Avg.	Std.	Min.	Max.
4	1164.2	1151.9	1.2	1150.9	1153.1	59.3	32.1	3.8	28.6	36.2
5	1133.6	1119.2	1.4	1118.0	1120.7	52.7	31.5	3.8	28.2	35.9
6	1080.5	1066.1	1.7	1064.6	1068.2	48.5	29.9	2.7	27.3	33.2
7	1071.1	1057.6	2.1	1055.7	1059.7	45.7	29.7	2.3	27.4	32.2
8	989.3	**978.2**	2.0	976.2	980.5	41.0	**27.4**	1.8	25.5	29.6
9	1010.6	997.0	2.6	994.5	1000.1	39.7	27.6	2.0	25.6	30.1
10	1011.3	997.5	2.7	995.1	1000.0	37.8	25.4	1.6	24.0	27.2
11	999.3	986.7	2.7	984.3	989.6	37.5	26.7	1.4	25.3	28.2
12	997.8	986.0	3.3	983.1	989.5	35.6	***25.0***	1.5	***23.5***	***26.7***
13	1005.3	994.8	3.1	991.9	998.4	36.5	27.6	1.3	26.4	29.2
14	992.6	981.2	3.0	978.4	984.8	***34.5***	26.1	1.4	24.7	27.8
15	995.7	985.4	3.6	982.0	989.2	38.4	31.6	1.1	30.5	32.8
16	***963.7***	***954.2***	3.6	***950.9***	***958.7***	36.0	29.5	1.0	28.5	30.6

(b) Berthing location pattern : ***Continuous***

# of partitions	Total service time (hours)					Yard block utilization (%)				
	SOBJ	MOBJ (Parete solutions)				SOBJ	MOBJ (Parete solutions)			
		Avg.	Std.	Min.	Max.		Avg.	Std.	Min.	Max.
4	1134.0	1122.3	1.2	1121.3	1123.5	55.2	30.9	3.4	27.8	34.8
5	1102.6	1089.8	1.5	1088.6	1091.4	52.1	31.3	2.9	28.7	34.5
6	1072.6	1057.8	2.3	1055.8	1060.3	50.1	30.1	2.6	27.6	33.0
7	1043.5	1030.1	1.9	1028.5	1032.2	45.8	30.2	2.5	27.9	33.0
8	1011.1	**997.4**	2.1	995.2	999.9	42.8	**28.8**	2.0	26.6	31.0
9	1013.1	999.7	3.0	996.9	1003.2	41.5	28.9	2.0	26.8	31.3
10	1001.6	988.3	3.2	985.3	991.8	37.9	26.7	1.8	24.9	28.7
11	1000.6	989.5	4.0	985.9	994.6	38.1	27.8	1.9	26.0	30.2
12	987.7	975.5	3.3	972.4	979.5	35.8	***25.6***	1.5	***24.1***	***27.4***
13	989.4	977.4	3.7	973.8	981.7	36.9	28.0	1.3	26.6	29.5
14	979.5	968.5	3.9	964.7	973.7	***34.2***	26.3	1.3	25.0	27.9
15	992.5	981.8	4.8	977.3	987.6	38.9	31.7	1.1	30.6	33.0
16	***959.3***	***949.6***	3.8	***946.0***	***954.2***	36.0	29.7	1.1	28.6	30.9

Note) **Under line** = total service time obtained by MOBJ in the problem with 8 partitions of yard block.
Bold & Italic = shortest total service time or smallest block utilization among the problem partitions of yard block.

7 CONCLUSION

We are concerned with the container storage planning problem at the terminal with various yard block sections, in order to carry out the container handling operations in the most efficient way. As a consequence of numerical experiments, the number of partitions defines the paths' location on perpendicular direction to quay length. Therefore, it is clear that the relationship between berthing location version and number of partitions effects on the yard trailers' routing, and that also does that on the total handling time and block utilization.

ACKNOWLEDGEMENTS

This research is partially supported by the MEXT under the Grant-in-Aid for Young Scientists-B Grant 23710169.

REFERENCES

Bazzazi, M., Safaei, N., Javadian, N. (2009), A genetic algorithm to solve the storage space allocation problem in a container terminal. Computers & Industrial Engineering, **36**, 1711-1725.

Glover, F., Laguna, M. (1997a), Chapter 1: Tabu search background, Tabu Search. Kluwer Academic Publishers, 1-16.

Glover, F., Laguna, M. (1997b), Chapter 4: Tabu search foundations: Longer term memory, Tabu Search. Kluwer Academic Publishers, 96-100.

Glover, F., Laguna, M. (1997c), Chapter 3: Tabu Search, Modern Heuristics Techniques for Combinatorial Problems, edited by Reeves, C.R., John Wiley & Sons, Inc., 70-141.

Han, Y., Lee, L.H., Chew, E.P., Tan, K.C. (2008), A yard storage for minimizing traffic congestion in a marine container transshipment hub, OR Spectrum, **30(4)**, 679-720.

Hansen, M.P. (1997), Tabu search for multiobjective optimization: MOTS, Proceedings of international Conference on Multiple Criteria Decision Making (MCDM'97).

Kim, K.H., Hong, G.P. (2006), A heuristic rule for relocating blocks, Computers & Operations Research, **33(4)**, 940-954.

Kim, K.H., Park, Y.M., Ryu, K.R. (2000), Deriving decision rules to locate export containers in container yards. European Journal of Operational Research, **124**, 89-101.

Kim, K.H., Park, K.T. (2003), A note on a dynamic space-allocation method for outbound containers. European Journal of Operational Research, **148**, 92-101.

Kozan, E., Preston, P. (1999), Genetic algorithms to scheduling container transfers at multimodal terminals. International Transactions in Operational Research, **6**, 311-329.

Lee, L.H., Chew, E.P., Tan, K.C., Han, Y. (2006), An optimization model for storage yard management in transshipment hubs, OR Spectrum, **28(4)**, 539-561.

Lim, A., Xu, Z. (2006), A critical-shaking neighborhood search for the yard allocation problem, European Journal of Operational Research, **174**, 1247-1259.

Nishimura, E. Imai, A., Janssens, G.K., Papadimitriou, S. (2009), Container storage and transshipment marine terminals, Transportation Research Part E, **45(5)**, 771-786.

Park, C., Seo, J. (2009), Mathematical modeling and solving procedure of the planar storage location assignment problem, Computers & Industrial Engineering, **57(3)**, 1062-1071.

Park, C., Seo, J. (2010), Comparing heuristic algorithms of the planar storage location assignment problem, Transportation Research Part E, **46(1)**, 171-185.

Yang, J.H., Kim, K.H. (2006), A grouped storage method for minimizing relocations in block stacking systems, Journal of Intelligent Manufacturing, **17(4)**, 453-463.

Zhang, C., Liu, J., Wan, Y.W., Murty, K.G., Linn, R.J. (2003), Storage space allocation in container terminals. Transportation Research Part B, **37**, 883-903.

THE CONTAINERSHIP FEEDER NETWORK DESIGN PROBLEM: THE NEW IZMIR PORT AS HUB IN THE BLACK SEA

Olcay Polat[1], Hans-Otto Günther[1] and Osman Kulak[2]

[1]*Technical University of Berlin, Department of Production Management, Straße des 17. Juni 135, 10623 Berlin, Germany, opolat@pau.edu.tr, hans-otto.guenther@tu-berlin.de*

[2]*Pamukkale University, Department of Industrial Engineering, Kinikli, 20100 Denizli, Turkey, okulak@pau.edu.tr*

ABSTRACT

Global containership liners design their transportation service as hub-and-spoke networks to increase the market linkages and reduce the average operational costs by using indirect connections. These indirect connections from the hub ports to the feeder ports called feeder networks are serviced by feeder ships. The feeder network design (FND) problem determines the smallest feeder ship fleet size with routes to minimize operational costs. Therefore, this problem could be described as capacitated vehicle routing problem with simultaneous pick-ups and deliveries with time limit. In our investigation, a perturbation based variable neighborhood search (PVNS) approach is developed to solve the FND problem which determines the fleet mix and sequence of port calls. The proposed model implementation has been tested using a case study from the Black Sea region with the new Izmir port (Candarli port) as hub. Moreover, a range of scenarios and parameter values are used in order to test the robustness of the approach through sensitivity analyses. Numerical results show that the new Izmir port has great potential as hub port in the Black Sea region.

Keywords: Maritime transport, feeder network design, variable neighborhood search

1 INTRODUCTION

The construction and introduction of mega containerships on the main international sea routes connecting major seaports gave rise to the need for containers to be stored up and distributed in a specific region. In conceptual terms, the feeder service is meant to collect containers from specific regions with feeder ships and feed large trunk container ships as to avoid their calling at too many ports (Multi Port Calling - MPC). It was the containership feeder shipping line that made the entire container service economically rational, efficient and more profitable, consequently cheaper and timely for the end users (Rudic and Hlaca 2005).

Regional feeder shipping lines have critical positions on the global hub-and-spoke (H&S) networks of shipping lines. Figure 1 shows two main feeder shipping systems: direct feeder shipping between hub and feeder port and indirect feeder shipping via line-bundling loops including more than one feeder port (Wijnolst et al. 2000). The first strategy has the lowest transit time but typically requires more feeders and smaller feeder containerships. Alternatively, indirect feeder shipping benefits from economies of feeder containership size, but incur longer distances and longer transit times. The feeder network comprises ships visiting a number of ports along the predefined lines of feeder ports in the region. The container feeder network design depends on the characteristics of feeder ships, characteristics of feeder ships ports, container demand and supply volumes of the ports and bunker costs as well as the operating/chartering/administration costs of the ships.

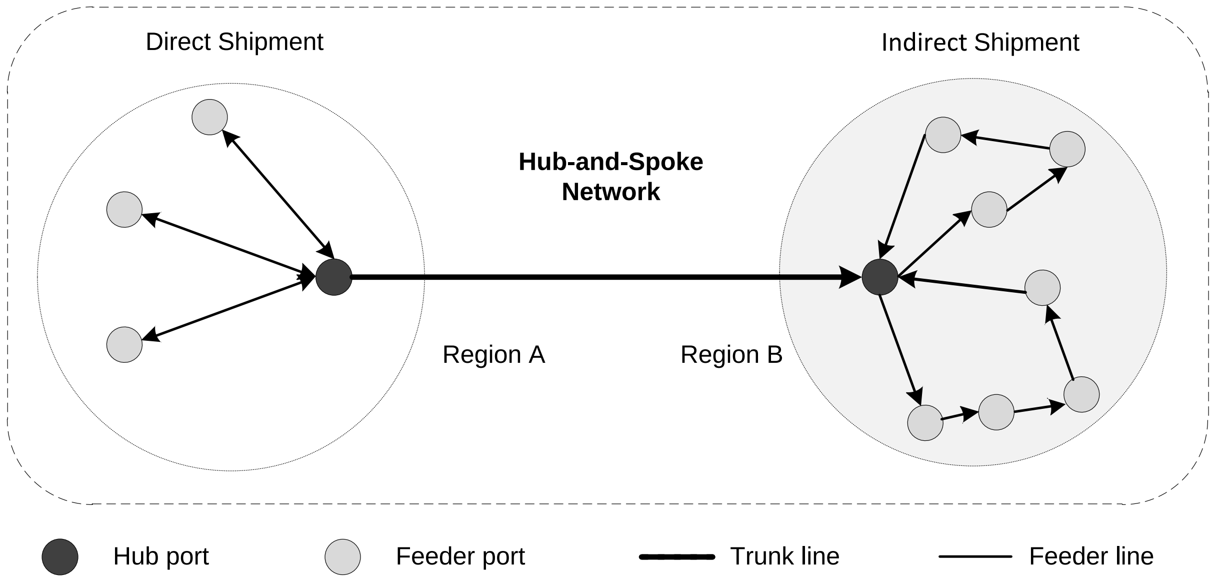


Figure 1. Feeder shipping networks as part of Hub-and-Spoke network

The problem considered is that of designing the network of indirect feeder containerships for feeder lines. In this problem assuming that a fleet of feeder capacitated container ships starting from the hub port would perform simultaneous container pickups and deliveries between hub and feeder ports under ship due date constraints for returning to the hub port at minimum cost for feeder containership liners.

Since container shipping involves considerable capital investments and huge daily operating costs, the appropriate containership feeder network design will affect the development of feeder containership liners. In this study, we focus on the potential hub role of a new port (Candarli) in the East Mediterranean and Black Sea region and develop an approach that deals with the feeder network design problem. The results are compared with current transshipment hub ports in the region. Due to the complexity of the problem, a perturbation based variable neighborhood search approach is proposed.

The remainder of this study is structured as follows. In the next section, a brief review of the relevant literature with a focus on liner network design is given. In Section 3, we present a model formulation for feeder network design problems. Next, a heuristic solution procedure is proposed. Section 5 summarizes the case study and service scenarios. Detailed numerical results are presented in Section 6. Finally, conclusions are drawn and suggestions for further research are given in Section 7.

2 LITERATURE REVIEW

Planning of container liner shipping operations has become a popular topic of academic research worldwide. Hence, a huge amount of papers has been published focusing on different planning aspects of container liner shipping (see Christiansen (2004), Notteboom (2004) and Kjeldsen (2008) for comprehensive reviews).

Only a few researches have been published which consider H&S operations with origin to destination (O-D) transportation processes as a whole. Takano and Arai (2009), Gelareh (2010), Gelareh and Nickel (2011) and Gelareh and Pisinger (2011) presented an approach for a H&S network with direct feeder services for container transportation. For a fixed number of hubs, their model determines the best network configuration of hub locations and direct shipment allocations for feeder ports that minimize the total costs of the system. In addition, Yang and Chen (2010) and Zacharioudakis et al. (2011) presented a genetic algorithm approach to optimize the combination of trunk and indirect feeder line networks for a shipping company. In the papers by Jin et al. (2005), Sun and Li (2006), Wang (2008), and Lu and Meng (2011), the authors proposed heuristic approaches to solve the containership routing problem with H&S operations by minimizing total costs.

For direct feeder shipping, Baird (2006) presented a methodology for evaluating and comparing hub ports in Northern Europe. Direct feeder shipping costs for current hub locations and a new proposed hub port in the Orkney Islands are compared. Ng and Kee (2008) evaluated optimal containership sizes of direct feeder services by using simulation models in Southeast Asia from the perspective of carriers.

Only a few researches have been published which consider the indirect feeder network design. Mourao et al. (2001) proposed an integer linear programming model for the assignment of ships to current indirect feeder routes. Catalani (2009) proposed a cost-minimization based expert system model for sequencing and scheduling of feeder ports for just one containership route in the Mediterranean area. Andersen (2010) proposed a mathematical model for service frequency requirements of predefined solid indirect liner feeder networks. The authors developed decomposition based heuristic approaches in order to solve the problem. Sambracos et al. (2004) presented a case study to dispatch small containers via coastal freight liners from a hub port to Greek island ports. Authors tried to minimize total operating cost including fuel consumption and port charges with a homogeneous ship fleet by meeting container shipment demand. Karlaftis et al. (2009) generalized a small container dispatching problem by minimizing total travel distance with simultaneous container pick-up and delivery operations and time deadlines constraints. They proposed a genetic algorithm (GA) based solution heuristic in order to solve the problem with soft time limits which tolerates violations of certain constraints.

3 THE FEEDER NETWORK DESIGN PROBLEM

The FND problem is given as follows. A set of feeder ports is located on a distribution network where feeder ports require both delivery and pickup operations. Each feeder port has to be served once for both operations with a given fleet of identical capacitated feeder ships. Each ship leaves the hub port carrying the total amount of containers it has to deliver and returns to the hub port carrying the total amount of containers it must pick-up. Each port (feeder/hub) also has a specified operation efficiency for loading and unloading containers to ships at the ports. The service time of the ports depends on port operation efficiency, ship sizes, the amount of loading and unloading containers and pilotage time for entering/exiting the port. Therefore, the total voyage duration of a ship is the sum of total travel time of the route and total service time of the hub and feeder ports. In order to determine the ship schedules and the staffing balance, each vessel has to finish its voyage before the maximal allowed duration is reached (the voyage starts in the hub port with commencing the loading operations to ships and completing the unloading operations from ships at eh hub port). Before starting a new voyage, each ship needs a lay-up interval for repair, cleaning, waste disposal etc. Total ship travel duration includes total voyage, lay-up and idle times. According to these considerations the FND has similarities with the *"vehicle routing problem with simultaneous pick-up and delivery with time limit"* (VRPSPDTL).

The FND problem aims to serve all contracted feeder ports by minimizing total operational costs in the planning period. For a feeder network provider, operational costs for the planning period include containership related fixed costs for the necessary number of ships (chartering/capital, operating, administration) and total service related variable costs (on sea bunker cost, on port bunker cost, port charges). Table 1 shows the related basic cost calculations.

Table 1. Basic calculations of total costs during the planning period

Parameter	Basic formulation
Total cost	Fix cost + Variable cost
Fix cost	Number of necessary ship ∗ (Chartering + Operating + Administration costs)
Variable cost	Number of service ∗ (Bunker (sea) + Bunker (port) + Port charges)
Number of necessary ship	ceil ((Voyage duration +Lay-up duration) / service frequency)
Number of service	Planning period / Service frequency
Voyage duration	On sea duration + On port duration (feeder) + On port duration (hub)
Idle duration	Number of necessary ship ∗ Service frequency − (Voyage + Lay-up duration)
Ship total duration	Voyage duration + Lay-up duration + Idle duration

Since our investigation is concerned with the design of a real world container feeder network, some assumptions have to be made in order to exclude elements of minor relevance and to focus on those aspects that are of paramount interest. Major assumptions of our model are the following: all parameter values are deterministic (no weather and seasonal effects), no direct delivery between feeder ports, queue time at ports is not considered, feeders' demand as well as feeders' container supply amounts cannot be divided, ship types are identical according to their carrying capacity, unlimited number of ships from each type, port handling and bunker costs are the same in all ports, there are no owned ships, fixed schedules and sailing frequencies for containerships are assumed. Vessel speed/fuel cost effect as well as straight/canal durations and costs are not considered. Container related costs are not included, since they have a given effect on the total cost.

4 THE PROPOSED METHODOLOGY

Exact methods for solving the FND problem are not practical for large problem instances because of the problem complexity. In this study, we therefore propose a perturbation based variable neighborhood search (PVNS) approach which applies the Savings Algorithm (SA) in order to gain a fast and effective initial solution. The PVNS is embedded with variable neighborhood search (VNS) to improve the initial solution by searching neighborhoods. In order to escape from local optima, an adaptive perturbation mechanism (APM) is developed.

The initial solution is constructed by means of the Savings Algorithm of Clarke and Wright (1964). This classic heuristic aims at merging sub-tours based on costs savings which can be achieved by combining two sub-tours to be served by one vehicle. In the literature, some enhancements of the Clarke and Wright savings algorithm have been suggested by adding new terms and parameterizing the savings formula. In this study, we use the savings formula proposed for the capacitated vehicle routing problems by Altinel and Öncan (2005).

Afterwards the initial solution is evaluated with a VNS improvement algorithm. The VNS, which is based on the idea of systematically changing the neighborhoods in order to improve the current situation, was introduced by Mladenović and Hansen (1997). VNS aims to explore the solution space which cannot be searched by local search. *Shaking*, *local search* and *move or not* operators are used in the implementation of the VNS. The shaking operator defines the search direction of the VNS by using the set of neighborhoods. The chance of reaching a global solution improves when combining the shaking operator with local search rather than using a single shaking operator. Therefore, each solution obtained through the shaking operator is used in the local search operator in order to explore promising new neighborhoods of the current solution. In this study we implemented the variable neighborhood descent (VND) algorithm as the local search operator. The VND aims to combine the set of neighborhoods in a deterministic way, since using more than one neighborhood structure could obtain a better solution (Hansen and Mladenović 2001).

In this study, a set of N [N1: 3-opt, N2: swap, N3: insertion, N4: 2-opt, N5: Exchange (m,n), N6: Cross, N7: Shift (0,1), N8: Replace (1,1)] neighborhood structures is employed in a deterministic order as shaking and local search operators. To avoid unnecessary movements, only feasible movements are admitted, i.e. those that do not violate the ship capacity and total duration limit of the route. Also, a reversed version of the routes which violate the vehicle capacity is applied, if the total delivery and pick-up amounts of the route are feasible.

The temporary solution which is obtained after the shaking and local search operators are applied is compared with the current solution in order to decide whether to *move or not.* In the proposed VNS and VND, the acceptance criterion of the temporary solution accepts only improvements. This procedure, however, could simply stick the search to a local optimum. Therefore, it is necessary to employ a strategy of accepting non-improving solutions. Perturbation is an effective strategy used to jump out of the local optimum and to search a new promising region. In this study, a novel perturbation method called adaptive perturbation mechanism (APM) is developed. This perturbation mechanism runs after a number of non-improving iterations counted from the last improving solution. In the APM, a set P_x [P_1: *double replace*, P_2: *double cross*, P_3: *triple shift*, P_4: *triple replace*, P_5: *triple cross*] perturbation structures is randomly run whenever the perturbation is called. In addition to the perturbation move, a local search with five previously defined intra-route neighborhood structures is applied in order to improve the perturbed solution quality, which is essential since a perturbation move satisfying the vehicle capacity and total route duration limits is always accepted. Moreover, violating moves are accepted only if the total route duration and ship capacity are below an acceptance limit (α). However, just one of the routes is allowed to use this violation and the travel duration of this route is punished with a very big penalty cost. This rule gives routes a potential improvement chance in the shaking and local search phase. The new developed perturbation structures for the APM are defined as follows:

Double Replace (P_1) is a combination of two times sequential *Replace (1,1). Double Cross* (P_2) is a combination of two times sequential *Cross* exchange. *Triple Shift* (P_3 is a combination of two times sequential *Shift (0,1)* movement between three routes. *Triple Replace* (P_4) is similar to *Triple Shift* by using the *Replace (1,1)* movement. *Triple Cross* (P_5) is similar to *Triple Shift* by using the *Cross* exchange structure.

5 CASE STUDY

The fact that the region is surrounded by several seas – the Black Sea, Mediterranean Sea, Adriatic Sea, Ionian Sea, Aegean Sea, and Marmara Sea – makes maritime shipping a prime area for growth going forward (see Figure 2). Container feeder shipping lines offer critical transport connections between the hinterland of this region and global trunk shipping lines. The feeder shipping dynamics of the region are mainly related to container transportation volumes of the trunk shipping lines between Far East and Europe. In recent years, parallel to the increase of container transportation volumes between Far East and Europe, an increase on the total container handling volume is observed in the regional feeder ports. The hub ports in the East Mediterranean area have a direct effect on the increasing importance of feeder lines in the region by serving as direct link to trunk lines. Thus feeder lines enhance the opportunity to attract more cargos in the region and ensure high capacity utilization (Varbanova 2011).

Turkey's ideal location between Asia and Europe gives its ports a competitive advantage and opportunity to develop into major transshipment hub ports. In this regard, Turkey has significant potential and several projects for the development of intermodal transport. One of these projects is the construction of a hub port in Izmir's Candarli district, in order to improve Turkey's hub port potential. In this region, the potential market areas of Candarli as a hub port could be categorized into four sub-regions: the Black Sea, the Sea of Marmara, the East Mediterranean sea and, the Aegean Sea.

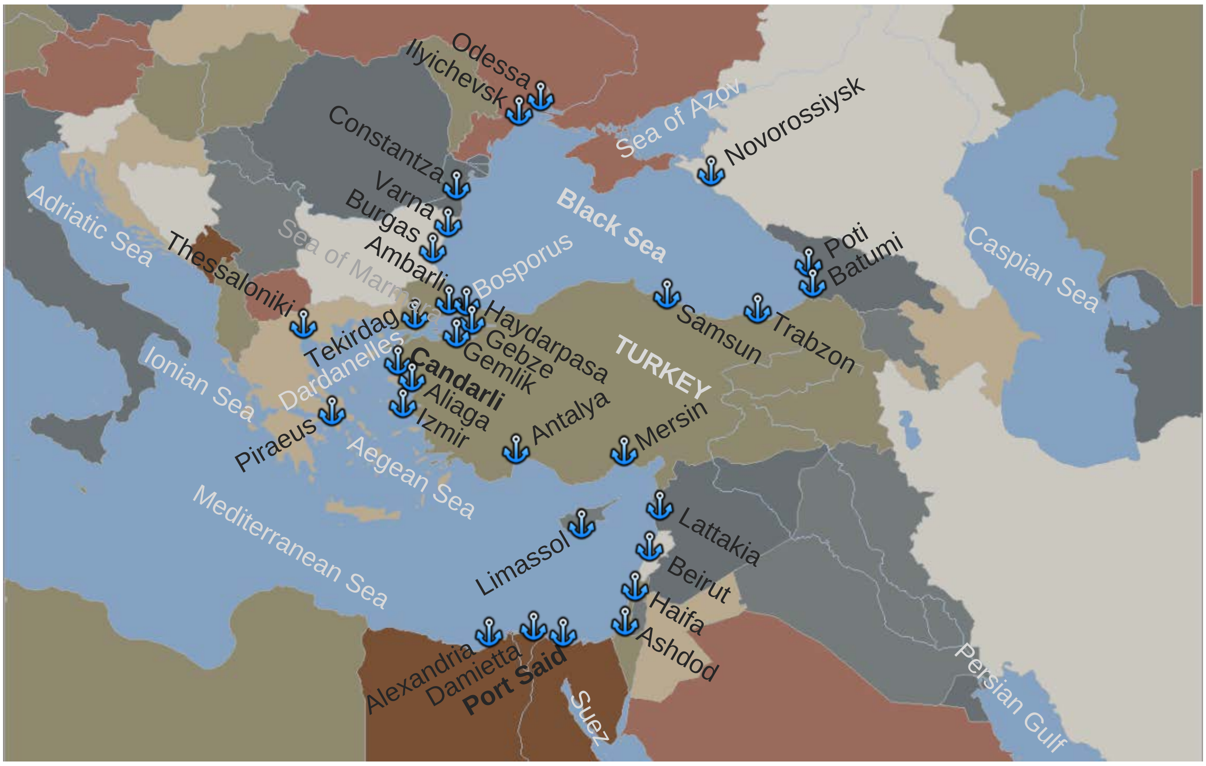


Figure 2. Regional feeder and hub ports

In this region, 38 container terminals at 28 feeder ports are served via a hub port for a feeder liner shipping company with 3680 TEU total daily demand and 2440 TEU total daily supply amount. The feeder liner currently designs its existing feeder network with a hub port of Port Said in North Egypt. However, after establishing Candarli as a new hub port alternative, feeder liner should reconsider its current feeder network. Therefore, in this study two different service scenarios are defined for the region. *The first scenario* is the current situation. Port Said serves as trunk hub port to feeder ports of the region. In *the second scenario* Candarli serves as trunk hub port to all feeder ports of the region for feeder liners. The scenarios are also tested under different time deadline and service frequency conditions for a 52 week planning period. The major cost items and ship costs for three ship types are shown in Table 2.

Table 2. Model costs and parameters

Parameter	Unit	Ship1	Ship 2	Ship 3
Capacity	TEU	4300	2600	1200
Operating speed	(knots)	22.60	19.90	17.40
Fuel consumption (on sea)	(tons/hour)	5.26	2.82	1.51
IFO 180 price (on sea)	($/ton)	647.50	647.50	647.50
Fuel consumption (on port)	(tons/hour)	0.26	0.14	0.08
MGO price (on port)	($/ton)	890.00	890.00	890.00
Charter cost	($/day)	12772.00	7579.00	5866.00
Operating costs	($/day)	6000.00	5707.00	4643.00
Administration cost	($/day)	552.00	3180.00	1380.00
Port charges	($/call)	35000.00	29000.00	22000.00
Handling cost (feeder port)	($/llift)	120.00	120.00	120.00
Handling cost (hub port)	($/llift)	120.00	120.00	120.00
Lay-up time (hub port)	(hour/call)	28.80	24.00	16.80
Pilotage time (all ports)	(hour/call)	2.00	1.80	1.50
Planning period	Days	364	364	364

Sources: Stopford (2009), VHSS (2012), BunkerIndex (2012)

6 NUMERICAL RESULTS

The proposed PVNS algorithm is coded using Matlab R2010b/Visual C# 4.0 and executed on an Intel Core 2 Duo T5750 2.0 GHz processor with 3 Gb RAM. As part of preliminary studies, experiments on the sequence of the shaking operators of the PVNS algorithm were conducted in order to determine the most effective sequence of the local neighborhood search set. The results demonstrated the effectiveness of the [N1: 3-opt, N2: Swap, N3: Insertion, N4: 2-opt, N5: Exchange (m,n), N6: Cross, N7: Shift (0,1), N8: Replace (1,1)] sequence. The same sequence is also used in the local search (VND) part of the VNS algorithm.

The perturbation mechanism is called after $1 * feeder\ port\ number$, i.e. 38, iterations counted from the last accepted move. The total route duration and the vehicle capacity violation acceptance limit (α) are used as 10%. This rule aims to allow customers to join another route for possible future improvements. The termination condition of the PVNS algorithm is used as maximum number of iterations between two improvements of the best solution. The termination condition is set to $100 * feeder\ port\ number$ iterations without improvement. The proposed PVNS algorithm is run ten times with different random seeds in order to measure its robustness. Table 3 shows the total costs for the current and alternate hub port under various service frequency and time deadline scenarios in the region.

Table 3. Scenario results for alternative hub port locations

Scenario	Hub	Frequency	Deadline	Total costs	Time
1	Port Said	7	2.5x7	286604.59	96.25
2	Port Said	7	3x7	288057.59	40.10
3	Port Said	7	3.5x7	286704.60	70.57
4	Port Said	7	4x7	**285125.22**	43.77
5	Port Said	7	4.5x7	288392.52	29.82
6	Port Said	3.5	2.5x7	332509.15	90.24
7	Port Said	3.5	3x7	332509.15	41.85
8	Port Said	3.5	3.5x7	330814.43	61.16
9	Port Said	3.5	4x7	331503.51	48.57
10	Port Said	3.5	4.5x7	333496.76	32.70
11	Candarli	7	2.5x7	259138.68	67.82
12	Candarli	7	3x7	**254338.80**	49.24
13	Candarli	7	3.5x7	257990.19	51.65
14	Candarli	7	4x7	257990.19	95.80
15	Candarli	7	4.5x7	258338.07	67.75
16	Candarli	3.5	2.5x7	300493.20	50.64
17	Candarli	3.5	3x7	299458.83	37.72
18	Candarli	3.5	3.5x7	299452.99	51.61
19	Candarli	3.5	4x7	296796.83	74.03
20	Candarli	3.5	4.5x7	299458.83	32.86

In Table 3, total costs include chartering costs, operating costs, administration costs, on-sea bunker costs, on-port bunker cost and port charges for a 52 week planning period. In the scenarios, the existing hub port (Port Said) presents minimum total operational costs of $285.125.220 with 7 days service frequency and 28 (4x7) days deadline for returning to the hub and finishing the unloading operations. The new proposed hub port (Candarli) presents minimum total operational cost of $254.338.800 with 7 days service frequency and 21 (3x7) days deadline. The network routes for the best scenarios for both hub port alternatives are shown in Figure 3. The proposed Candarli port shows around 12% cost advantage compared to the existing hub port of the network. Feeder and trunk shipping lines could transfer their transshipment operations to Candarli, as long as Candarli port authorities keep their container handling costs and relevant service quality at a favourable level.

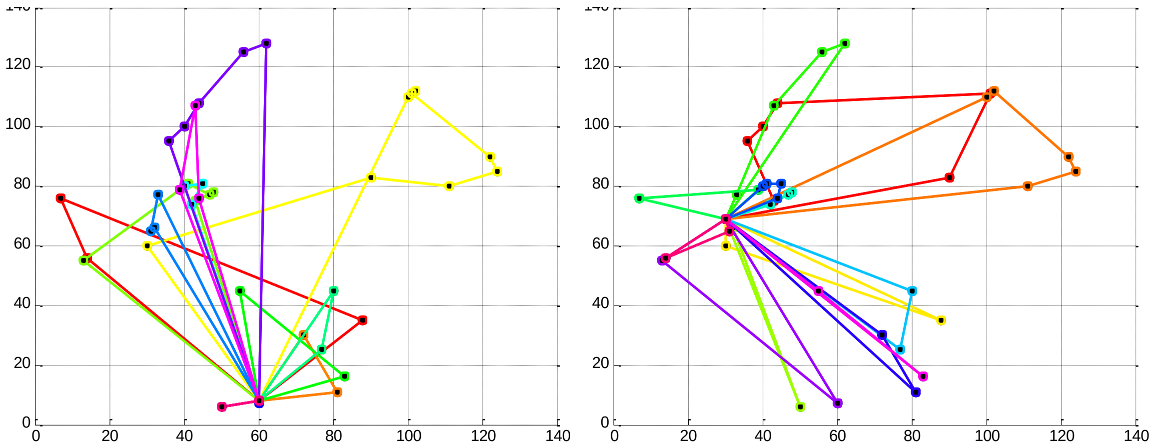


Figure 3. Feeder routing networks for Port Said and Candarli port

Table 4 presents a comparison between costs, service and duration rates of alternative hub ports. As in trunk shipping, feeder shipment is highly sensitive to bunker fuel costs as they represent between 24.98 and 27.96% of total operational cost. However bunker costs contain almost 40-45% of total operational costs for trunk shipping lines. Since total voyage distances of feeder networks are less than those of trunk networks, total network costs contain more ship based fixed costs such as chartering, operating and administration. Therefore ship type selection of the feeder networks are more fixed cost oriented. Since Candarli port has shorter distance to feeder ports, this port based feeder network selects relatively small containerships. On the other hand, the Port Said based feeder network creates its routing network with mid-sized containerships. 4600 TEU containerships are not appropriate for both hub alternatives because of its relatively high fixed costs. Still, from a comparative perspective feeder shipping liners' ship selection is sensitive to fuel price and network distance.

Table 4. Cost rates for Port Said and Candarli

	Parameter	Port Said	Candarli
Model Cost	Total cost ($1000)	285164.07	254407.03
	Chartering cost	21.81%	23.22%
	Operating cost	16.48%	17.72%
	Administration cost	8.89%	8.66%
	Bunker cost (on sea)	27.96%	24.98%
	Bunker cost (on port)	5.41%	5.46%
	Port charges	19.45%	19.96%
Ship	Number of routes	12	13
	Total necassary ship	23	23
	1200 TEU	8.70%	30.43%
	2600 TEU	91.30%	69.57%
	4300 TEU	0.00%	0.00%
Avg. Duration	Total duration (Hour)	322.00	297.23
	On sea duration	23.39%	21.23%
	Port duration (feeder)	40.28%	40.18%
	Port duration (hub)	22.75%	22.77%
	Lay-up duration	7.27%	7.14%
	Idle duration	6.31%	8.69%

It could be expected that as long as the network distance is enlarged, the selected ship capacities will increase in order to meet the balance between fixed and variable costs. Average ship duration for the Candarli related feeder network is about 297.23 hours. The most significant durations are related to feeder port service (40.18%) and hub port service (22.77%). As it expected, the basic duration variance between the alternative hub ports are on the see voyage durations (23.39% and 21.23%), which is the natural effect of geographical difference between the ports.

7 CONCLUSION

In this study, we focus on the potential hub role of a new port (Candarli) in the East Mediterranean and Black Sea region and develop an approach that deals with the feeder network design problem. According to the demand distribution, this study is to determine the feeder network, fleet mix, time deadlines and service frequencies by obtaining the minimum operational costs. Therefore, we proposed a novel hybrid search method called perturbation based variable neighborhood search (PVNS) to solve the feeder containership network design (FND) problem. PVNS is based on the Savings Algorithm (SA), variable neighborhood search (VNS) and adaptive perturbation mechanism (APM). We used eight local neighborhood search structures as shaking and local search operators of the VNS algorithm. A variable neighborhood descent (VND) procedure is used to perform the local search. We use five adaptive perturbation structures in order to escape from local optima. The total operational costs of the optimal feeder networks of existing and alternate hubs are calculated and compared. From the numerical results it can be concluded that Candarli has great market advantage as long as port authorities keep their container handling costs and relevant service quality at a favourable level. The study could be extended by considering cost and durations differences between feeder and trunk shipping networks.

REFERENCES

Altınel K, Öncan T (2005) A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. Journal of the Operational Research Society 56:954–961

Andersen MW (2010) Service Network Design and Management in Liner Container Shipping Applications (Chapter 5). Ph. D. thesis, Technical University of Denmark,

Baird AJ (2006) Optimising the container transhipment hub location in northern Europe. Journal of Transport Geography 14:195-214

BunkerIndex (2012) South Europe Bunker Prices. http://www.bunkerindex.com/prices/seurope.php. Accessed 15.06.2012

Catalani M (2009) Ship scheduling and routing optimization: An application to Western Mediterranean area. European Transport 42:67-82

Christiansen M, Fagerholt, K., Ronen, D. (2004) Ship routing and scheduling: Status and perspective. Transportation Science 38 (1):1-18

Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12:568-581

Gelareh S, Nickel S (2011) Hub location problems in transportation networks. Transportation Research Part E 47:1092-1111

Gelareh S, Nickel S, Pisinger D (2010) Liner shipping hub network design in a competitive environment. Transportation Research Part E 46:991-1004

Gelareh S, Pisinger D (2011) Fleet deployment, network design and hub location of liner shipping companies. Transportation Research Part E 47:947-964

Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. European Journal of Operational Research 130:449-467

Jin Z-H, You Q-H, Wang J (2005) The container ship routing problem and its practical solutions. Paper presented at the International Conference on Management Science and Engineering Incheon/Korea

Karlaftis MG, Kepaptsoglou K, Sambracos E (2009) Containership routing with time deadlines and simultaneous deliveries and pick-ups. Transportation Research Part E 45 (1):210-221

Kjeldsen KH (2008) Classification of routing and scheduling problems in liner shipping. Centre for Operations Research Applications in Logistics (CORAL), orking paper series L-2008-06:(p25)

Lu K, Meng Q (2011) Containership routing problem with hub-and-spoke operations. Paper presented at the Transportation Research Board 90th Annual Meeting, Washington/USA,

Mladenović N, Hansen P (1997) Variable neighborhood search. Computers & Operations Research 24:1097-1100

Mourao MC, Pato MV, Paixao AC (2001) Ship assignment with hub and spoke constraints. Maritime Policy & Management 29 (2):135-150

Ng AKY, Kee JKY (2008) The optimal ship sizes of container liner feeder services in Southeast Asia: A ship operator's perspective. Maritime Policy & Management 35 (4):353-376

Notteboom TE (2004) Container Shipping And Ports: An Overview. Review of Network Economics 3 (2):86-106

Rudic D, Hlaca B (2005) Feeder service to promote revitalization of the container transport in the Adriatic ports. Naše more, Journal of marine science and technology 52 (1-2):37-44

Sambracos E, Paravantis JA, Tarantilis CD, Kiranoudis CT (2004) Dispatching of small containers via coastal freight liners: The case of the Aegean Sea. European Journal of Operational Research 152:365-381

Stopford M (2009) Maritime Economics. Third Edition. Routledge, New York

Sun X, Li N (2006) A new immune genetic algorithm for the large scale ship routing problems. Paper presented at the International Conference on Management of Logistics and Supply Chain Chang Sha/Chine - Sydney/Australia,

Takano K, Arai M (2009) A genetic algorithm for the hub-and-spoke problem applied to containerized cargo transport. Journal of Marine Science and Technology 14:256-274

Varbanova A (2011) Current issues in operational planning of general cargo transportation on container feeder lines in the Black Sea region. The international virtual journal for science, techniques and innovations for the industry "Machines, Technologies, Materials” (3):35-38

VHSS (2012) Containership Time-Charter-Rates. http://www.vhss.de/contex.php. Accessed 15.06.2012

Wang C-x (2008) Optimization of hub-and-spoke two-stage logistics network in regional port cluster Systems Engineering - Theory & Practice 28 (9):152-158

Wijnolst N, Waals F, Bello F, Gendronneau Y, Kempen Dv (2000) Malacca Max (2) Container Shipping Network Economy. Delft University Press, Delft/Netherlands

Yang Z, Chen K (2010) Optimization of shipping network of trunk and feeder lines for inter-regional and intra-regional container transport Journal of the Eastern Asia Society for Transportation Studies 8:682-693

Zacharioudakis PG, Iordanis S, Lyridis DV, Psaraftis HN (2011) Liner shipping cycle cost modelling, fleet deployment optimization and what-if analysis. Maritime Economics & Logistics 13:278-297

HYBRID BACTERIAL FORAGING OPTIMIZATION APPROACH FOR DESIGN OF PID CONTROLLER IN MAGLEV TRANSPORTATION SYSTEM

Jae-Hoon Cho[1] and Yong-Tae Kim[2]

[1]*Smart Logistics Technology Institute, Hankyong National University, Korea,*
jhcho@hknu.ac.kr

[2]*Dept.of Information & Control Engineering, Hankyong National University, Korea*
ytkim@hknu.ac.kr

ABSTRACT

This paper presents a hybrid bacterial foraging optimization method for the design of a proportional-integral-derivative(PID)-based levitation controller for a magnetic levitation(Maglev) transportation system with an electromagnetic suspension system(EMS). Since the EMS-type Maglev system is mainly loop unstable, highly nonlinear, and has time-varying parameters, an effective stable controller is required the suitable performance. In this paper, a hybrid bacterial foraging optimization with a mutation-based queen-bee optimization method was used to optimize the PID controller parameters of an EMS-type Maglev system. The effectiveness of the proposed method was verified by numerical simulations and the simulation results showed that the proposed method is more efficient than conventional intelligent methods.

Keywords: Magnetic levitation, PID controller, Bacterial foraging optimization, Queen-bee optimization

1 INTRODUCTION

Recently, many magnetic-levitation (Maglev) transportation systems have been constructed, tested, and improved because they have several advantages that are attributed to a noncontact system, which reduces noise, component wear, vibration, maintenance costs, etc. (Kim *et al* 2006). In general, the levitation method of a Maglev transportation system can be divided into electro-magnetic suspension(EMS) and electro-dynamic suspension (EDS). The EMS system uses attractive force to suspend objects, and the EDS system uses the repulsive force to support objects. Since the EDS system requires complicated hardware such as super-conducting magnets or permanent magnets, it is technically difficult and expensive to implement. The EMS system is intrinsically unstable and requires a delicate stabilizing control(Wai *et al* 2011). To control of an EMS, a PID controller and an observer-based state-feedback controller are conventionally used (Park *et al* 2008).

However, conventional PID controllers have been widely used in industry because of their simple control structure, easy design, and inexpensive cost. Although a conventional PID controller cannot provide perfect control performance if the controlled phenomenon is highly nonlinear and uncertain, PID controllers are still used in many real world control applications. Over the years, several typical methods have been proposed for the design of PID controllers such as the Ziegler-Nichols method, the frequency-domain method, and the time-domain method. With the development of computer technology and artificial intelligence (AI), PID controller designs based on intelligent optimization algorithms such as genetic algorithm (GA) (Kwok, D.P et al 1992) and particle swarm optimization(PSO) (Gaing 2004) have been proposed. However, these algorithms have still some shortcomings such as premature convergence and uncertain performance. In order to overcome these disadvantages, many methods have been proposed in order to improve the performance of genetic algorithms(Srinivas, M. et al 1994; Smith, J.E. et al 1997). Jung(2003) proposed a new evolution method called queen-bee evolution in order to enhance the optimization capability of genetic algorithms. This algorithm is analogous to nature where the queen-bee plays a major role in the reproduction process. Queen-bee evolution facilitates genetic algorithms to quickly approach global optimum while decreasing the probability of premature convergence.

Natural selection tends to eliminate animals with poor foraging strategies and methods for locating, handling, and ingesting food, and it favors the propagation of genes of those animals that have successful foraging strategies. Based on this concept, Passino(2002) proposed an optimization technique known as bacterial foraging optimization (BFO). The BFO algorithm mimicking biological bacterial food-searching behavior has been applied in several optimization problems, and it can be categorized based on the chemotaxis algorithm and bacterial foraging algorithm. The chemotaxis algorithm was proposed as an analogy to the way bacteria react to chemoattractants in concentration gradients. The bacterial foraging algorithm is based on bacterial chemotaxis, reproduction and elimination-dispersal events (Kim et al 2007). Although BFO has been efficiently applied to various fields, it has some problems related to selection of parameters such as elimination-dispersal probability and reproduction size.

This paper introduces an approach for the design of a PID controller using a hybrid optimization algorithm with the advantages of both of bacterial foraging optimization and queen-bee algorithm. Simulation results are provided to show that the proposed method is more effective than conventional intelligent optimization methods.

2 EMS-TYPE MAGLEV SYSTEM

Many studies dealing with magnetic levitation system modeling are based on model linearization using a Taylor series. In this paper, the linearized Maglev model based on a Taylor series of the actual nonlinear dynamic model and force distribution at nominal operating points is used.

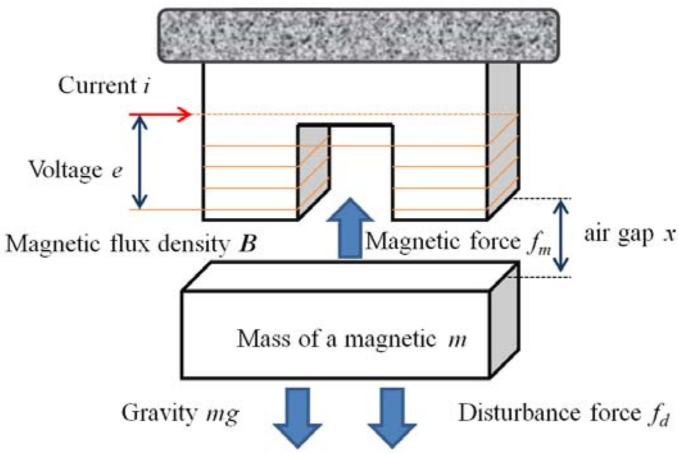


Figure 1. Structure of magnetic levitation system

Figure 1 shows the simplified EMS-type Maglev system used in this paper. The dynamic equation of a Maglev system is as follow:

$$m\frac{d^2x}{dt^2} = mg - f_m + f_d = mg - k_f\left(\frac{i}{x+X_0}\right)^2 + f_d \tag{1}$$

where $k_f = N^2\mu_0 S/4$; n is the number of winding turns of the electromagnet; μ_0 is the permeability of free space; S is the pole face area of the electromagnet; and x, f_m and $f_d(t)$ denote the air gap, magnetic force and disturbance force input, respectively. If R is total the resistance of the circuit, then an instantaneous voltage e can be described by

$$e = \frac{d}{dt}(Li) - Ri = -\frac{Q}{(x+X_0)^2} i \frac{dx}{dt} + L\frac{di}{dt} + Ri_c \tag{2}$$

$$X_0 = \frac{l_m}{2\mu_s}, \quad Q = \frac{\mu_0 N^2 S}{2} \tag{3}$$

where l_m is the magnet length of the guideway and iron core along the magnetization direction. By using linear approximations of the attraction force for excursions around the nominal equilibrium point and the system parameters of the Maglev system in Table 1, we can obtain the transfer function of the open-loop system of a linear model as follows (Kim 2005):

$$G(s) = \frac{-1419.60}{s^3 + 283.50s^2 3 + 392.38s - 551880} \tag{4}$$

Table 1. System parameters of the Maglev transportation system

Parameters		Nominal value
$Q[Hm]$	Speed electromotive force	13.1812×10^{-4}
$k[Hm]$	Force coefficient	6.5906×10^{-4}
$L[H]$	Nominal inductance	0.1097
$R[\Omega]$	Coil resistance	31.1
$X_1[m]$	$X+X_0$	0.01
$m[kg]$	Magnet mass	0.01058
$I[A]$	Nominal current	0.125

3 OPTIMIZATION ALGORITHMS INSPIRED BY NATURAL SELECTION

3.1 Bacterial Foraging Optimization

To apply BFO to an optimization problem, we describe a conventional BFO as follow (Passino 2002). In the minimal problem, the main goal of the BFO-based algorithm is to find the minimum of $J(\theta)$ $\theta \in \Re^p$, not the gradient $\nabla J(\theta)$. Here, θ is the position of a bacterium, p is dimension of the search space , and $J(\theta)$ denotes an attractant-repellant profile, i.e., at the location of the nutrients and noxious substances, $J(\theta) < 0$, $J(\theta) = 0$, $J(\theta) > 0$ represent the presence of nutrients, a neutral medium, and noxious substances, respectively. The population of bacteria can be defined by

$$P(j,k,l) = \{\theta^i(j,k,l) \mid i = 1,2,\cdots,S\} \tag{5}$$

where $\theta^i(i,j,k)$ represents the position of each member in the population of the bacteria S at the j^{th} chemotactic step, the k^{th} reproduction step, and the l^{th} elimination-dispersal event. Let $J(i,j,k,l)$ denote the cost at the location of the i^{th} bacterium $\theta^i(j,k,l) \in \Re^p$, and the bacterial position after the next chemotactic step can be represented by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j) \tag{6}$$

where $C(i) > 0$ is the size of the step taken in the random direction $\phi(j)$ specified by the tumble. If the cost $J(i,j+1,k,l)$ at $\theta^i(j+1,k,l)$ is greater than at $\theta^i(j,k,l)$, then another chemotactic step of size $C(i)$ in this same direction will be taken and repeated up to a maximum number of steps N_s , which is the lifetime of the bacteria, as long as it continues to reduce the cost. During the process of chemotaxis, the bacterium that has searched for the optimal position tries to provide an attractant or repellent signal for the swarm behaviors of a group.

3.2 Queen-bee Evolution for Genetic Algorithms

To enhance the search capability of genetic algorithms, a novel evolution method called queen-bee evolution was introduced by Jung(2003). Queen-bee evolution is one of the evolution methods inspired by bee behavior. In the queen-bee algorithm, the two probabilities of mutation, normal and strong, are used to enhance exploitation and exploration of the genetic algorithm.

A normal mutation probability increases the exploitation of genetic algorithms. However, it also increases the probability that genetic algorithms fall into premature convergence, and these results in a decrease in the performance of genetic algorithms. To decrease the probability of premature convergence, some individuals in queen-bee evolution are strongly mutated by the strong mutation probability. This reinforces the exploration of genetic algorithms. These two probabilities enable genetic algorithms to evolve quickly as well as to maintain good solutions.

4 DESIGN OF PID CONTROLLER FOR MAGLEV SYSTEM BY HYBRID BACTERIAL FORAGING OPTIMIZATION

4.1 Object Function for Hybrid Bacterial Foraging Optimization

In the design method of a PID controller using intelligent optimization algorithms, the selection of the object or cost function is important to achieve the performance desired by the user. In general, a PID controller design method using performance indexes such as integrated absolute error (IAE), integral of squared-error (ISE), or integrated time-weighted- absolute -error (ITAE) is often employed in control system design because it can be evaluated analytically in the frequency domain. In this paper, we use the ITAE method shown in Eq. (7) to estimate the performance of the PID controller designed by the proposed algorithm, because this criterion balances error size and duration and avoids positive and negative errors cancelling each other (da Silva et al 2000).

$$ITAE = \int_0^\infty \left| e^2(t) \right| dt \tag{7}$$

4.2 Hybrid bacterial foraging optimization

The proposed hybrid BFO procedure can be briefly described as follows. This algorithm consists of two major steps, namely a bacterial foraging step and a queen-bee step. The queen-bee step is executed in the next chemotactic step to improve the performance of BFO. In the present paper, the best bacterium in the chemotactic step is the selected as queen bee I_q , and the parent bees I_m are chosen by the *Roulette selection* algorithm. Moreover, a simplified process without the reproduction step and the elimination-dispersal event is used for fast computation time.

[step 1] Initialize parameters p, S, N_c, $C(i)\,(i=1,2,\ldots,N)$,.

[step 2] Chemotaxis loop: $j = j+1$

[substep a] For *i =1,2,…,S,* take a chemotactic step for bacterium *i* as follows.

[substep b] Compute *J*(*i*,*j*) by (7)

[**Queen-bee step**]

The best bacterium is chosen as I_q.

select *P*(*t*) from *S*(*i*)

$P(t) = \{ (I_q(t\text{-}1), I_m(t\text{-}1))$

do mutation (*)

for *i*=l to *n*

if $i \le (\xi \times n)$: ξ is mutation rate

do mutation with normal mutation probability p_m

else

do mutation with strong mutation probability p_{sm}

end if

end for

evaluate *P*(*t*) and update *J*(*i*,*j*)

[**Queen-bee end**]

[substep c] Let $OF_{last} = P(i,j)$ to save this value since we may find a better object value via a run.

[substep d] Tumble: generate a random vector $\Delta(i) \in \Re^p$

[substep e] Update location of each bacterium by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{8}$$

[substep f] Swimming: Let *m*=0(counter for swim length)

a) While $m < N_s$

b) If $J(i, j+1) < J_{last}$ (if doing better), let $J_{last} = J(i, j+1)$

c) Else, let $m = N_s$. This is the end of the while statement.

d) Go to next bacterium $(i+1)$ (i.e., go to [substep b])

[step 3] If $j < N_c$, go to [step 2]; (In this case, continue chemotaxis, since the life of the bacteria is not over) otherwise end.

5 SIMULATION AND RESULTS

To verify the proposed design method, a Maglev model was established using *MATLAB*. Table 2 lists initial parameters for the proposed algorithm. Eq. 9 is the PID controller model used in the present paper, and the initial range of parameters of the PID controllers is $z_1 \in [0,50]$, $z_2 \in [0,50]$, $K \in [-100,0]$.

$$K_c(s) = \frac{K(s+z_1)(s+z_2)}{s} \tag{9}$$

Table 2. Initial parameters of the proposed method

	Parameter	Value
S	The initial bacteria population size for BPO	100
Nc	Chemotactic steps	300
Ped	Elimination-dispersal with probability,	0.8
C	The step size for a swimming bacteria	4
Pop	Population size for Queen-bee	20
ξ	Normal mutation rate	$0.4 \le \xi \le 0.8$
P_m	normal mutation probability	0.05
P_{sm}	strong mutation probability	$0.6 \le \xi \le 1.0$

Table 3 lists the simulation results of the proposed method obtained by variable mutation rate and strong mutation probability. Figure 2 shows the step responses of a Maglev transportation system with the PID parameters selected by each algorithm using the integrated time weighted absolute error. The simulation results of each algorithm are summarized in Table 4. From Tables 3 and 4, we can see that the proposed algorithm obtains better performance than the others.

Table 3. Simulation results according to variable strong mutation probability and mutation rate

ξ	P_{sm}	$\times$ e-4	Z1	Z2	K
	0.6	1.1149	49.51	48.56	-99.96
0.4	0.8	1.1149	48.85	49.33	-99.95
	1.0	1.1148	49.08	48.74	-99.97
	0.6	1.1150	49.17	48.41	-99.94
0.6	0.8	1.1150	48.44	49.15	-99.95
	1.0	1.1150	49.12	48.53	-99.93
	0.6	1.1152	48.87	48.84	-99.89
0.8	0.8	1.1148	49.01	48.98	-99.95
	1.0	**1.1146**	49.26	48.75	-99.99

Table 4. Simulation results for each algorithm

Algorithm	ITAE ($\times$ e-4)	Z1	Z2	K
GA	1.1199	48.80	49.13	-99.09
PSO	1.1195	46.64	49.78	-100.00
HBF	**1.1146**	49.26	48.75	-99.99

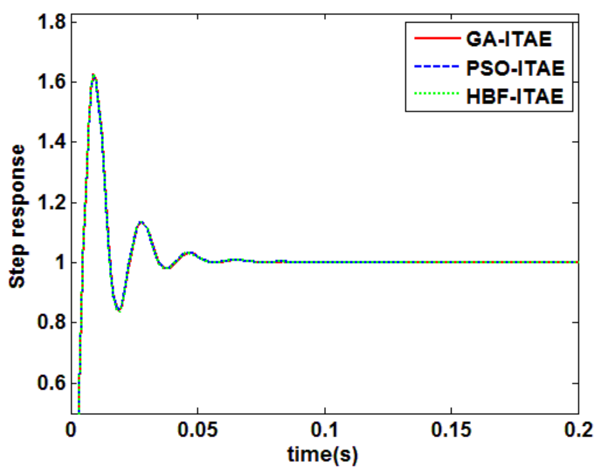


Figure 2. Step responses of Maglev transportation system by each algorithm using ITAE

6 CONCLUSION

In this paper, a hybrid bacterial foraging optimization method for a PID-based levitation controller design was proposed. This algorithm consists of two major steps including bacterial foraging step and a queen-bee step. To search for the optimum PID parameters of a Maglev transportation system, the bacterial foraging steps carry out exportation by random direction, and the queen bee steps perform the exploitation to improve the performance of the parameters selected by the bacterial foraging steps.

To verify the proposed design method, a simple Maglev system based on MATLAB was established and evaluated. Simulation results are provided to show that the proposed method is more effective than conventional intelligent methods.

ACKNOWLEDGMENT

This work was supported by the GRRC program of Gyeonggi Province [(GRRC-Hankyong 2011-B01), Research Center for Smart Logistics Technology]

REFERENCES

Lee, H.W., Kim, K. C. and Lee, J. (2006), Review of maglev train technologies, *IEEE Trans. Magn.*, **42**, 1917-1925.

Wai, R.J., Lee, J.D. and Chuang, K.L. (2011), Real-Time PID control strategy for Maglev transportation system via Particle Swarm Optimization, *IEEE Trans. Ind. Electron.*, **58**, 629-646.

Park, J.H. and Baek, Y. S.(2008), Design and analysis of a maglev planar transportation vehicle, *IEEE Trans. Magn.*, **44**, 1830-1836.

Kwok, D.P. and Wang, P.(1992), Fine-tuning of Classical PID controllers based on genetic algorithms, *IEEE International Workshop on Emerging Technologies and Factory Automation*, 37 -43.

Gaing Z. L.(2004), A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Trans. Energy Conver.*, **19**, 384 – 391.

Srinivas, M. and Patnaik, L.M.(1994), Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Trans. Syst. Man, Cybern*., **24**, 68-67.

Smith, J.E. and Focarty, T.C.(1997), Operator and parameter adaptation in genetic algorithms, *Soft computing a fusion of foundations, methodologies an applications*, **92**, 81-87.

Jung, S.H. (2003), Queen-bee evolution for genetic algorithms, *Electron. Letters*, **39**, 575-576.

Passino, K.M. (2002), Biomimicry of bacterial foraging for distributed optimization and control, *IEEE contr. Syst. Mag.*, **22,** 52-67.

Kim, D. H., Abraham, A. and Cho, J. H.(2007), A hybrid genetic algorithm and bacterial foraging approach for global optimization, *Inform. Sciences*, **177,** 3918-3937.

Kim, J.M. (2005), A Design of Optimal PID controller to a Magnetic Levitation System Using RCGA, *KSPSE fall conference*, 199-205.

da Silva, W.G., Acarnley, P.P. and Finch, J.W.(2000), Application of genetic algorithms to the online tuning of electric drive speed controllers, *IEEE Trans. Ind. Electron.*,**47,** 217-219.

OPTIMIZING SECURITY MANPOWER SCHEDULING WITH MULTIPLE SECURITY LEVELS AT AIRPORTS

Sangwon Seo[1] and Chulung Lee[2,†]

[1] *Graduate School of Information Management and Security, Korea University, Seongbuk-gu, Seoul 136-701, KOREA, sayst@korea.ac.kr*

[2] *Division of Industrial Management Engineering and Graduate School of Management of Technology, Korea University, Seongbuk-gu, Seoul 136-701, KOREA, leecu@korea.ac.kr*

ABSTRACT

In this paper, we developed an integer programming model and implemented a multi-objective genetic algorithm (MOGA) for security manpower scheduling problem arising at airports. Airport security manpower scheduling problem is to decide how many security checkpoints to operate in each time period and how to allocate security teams into the required security checkpoints in order to satisfy operational constraints such as passenger demands, work hours limits, and security levels. In order to validate the proposed approach, we conduct a case study to test the efficiency of proposed methods based on real data obtained from the airport. The results show that the integer programming model always confirms an optimal solution and the MOGA approach also generates an optimal solution or very close to that of the optimal solution within a short amount of time. For further extension, this paper proposes the passenger monitoring methods to reduce overall passenger waiting time and the performance is validated with the developed simulation model. The results of simulation shows that the proposed passenger monitoring method helps to improve security line balance problem, and its effectiveness works better when the expected passenger waiting time is longer than usual.

Keywords: Manpower scheduling, Airports security, Integer programming, Multi-objective genetic algorithm

1 INTRODUCTION

Airport security problem received greater attention from the aviation industry after the terrorist attacks of September 11, 2001. Since then, the overall security processes became much more complicated and intense, which significantly increases passenger waiting times at security checkpoints in the post 9.11 era. Therefore, many air passengers nowadays feel stressed from the long waiting lines and pat down searches at security checkpoints. Also, excessive passenger waiting time often recognized as lost revenue for airports authority since they lose opportunities to increase non-aeronautical revenues from commercial facilities at airport. In order to solve these issues, the better utilization of airport security checkpoints and manpower scheduling is required. Unlike other manpower application problems, airport security manpower scheduling problem deal with multiple security levels. Airport security levels are a crucial factor for estimating required team for each time period since the passenger screening time varies with different security levels. Also, we consider all the possible shifts combinations and the combinations of the problem make hard to solve in a short time. Thus, this paper propose effective methods to minimize passenger waiting time

[†] Corresponding author: e-mail : leecu@korea.ac.kr, Tel : 82-2-3290-3395, Fax : 82-2-929-5888

while satisfying passenger demand and considering the total operating costs and work fairness based on an integer programming and a multi-objective genetic algorithm (MOGA).

2 LITRATURE REVIEW

This paper review relevant literatures for manpower scheduling problem related with aviation industry. Holloran and Byrn (1986) developed a computerized station manpower planning system for United Airlines. This system is developed by integer and linear programming with network optimization techniques. Schindler and Semmel (1993) developed modified set covering integer linear programming to solve staffing problem at Pan Am. The proposed mathematical model considers part time employees, however, only can handle a single typical day. Similarly, Brusco and Jacobs (1995) developed and implemented two modules based on column generation approach for enhancing the tour-scheduling process associated with United Airline's manpower planning system called Pegasys. Then, a local search heuristics based on simulated annealing (SA) approach has introduced. hRecently, Yang *et al.* (2003) considered the problem of airline maintenance manpower planning. The problem is solved by a mixed integer programming model with the objective of minimizing the total number of man-hours. The problem includes flexible strategies such as changing the number of shift and squad members, and employee's working hours. Ernst *et al.* (2004) addressed a comprehensive literature review on manpower scheduling problem which applied to various applications including transportation systems, call centers, health care systems, and so on. Besides manpower scheduling problem, few researches deal with scheduling problem regarding airlines problem. Huang *et al.* (2006) studied workload balancing problem arising at air cargo terminal and develop a stochastic mixed integer linear program model with an accelerated Benders decomposition algorithm. Lee *et al.* (2007) developed a multi-objective genetic algorithm for robust flight scheduling problem. Then, the performance of the model is evaluated with a simulation model. Rong and Grunow (2009) developed an integrated mixed integer linear programming model for determining manpower requirements at the air cargo terminal. The objective function is to minimize manpower costs over planning horizon. They developed integrated approach and compared the results with a two-stage approach. The results of the proposed integrated approach give better solutions. Stolletz (2010) developed a binary integer programming model with flexible employee contracts for check-in counters at airports. The objective function is to minimize the costs of all assigned shifts. Recently, Seo *et al.* (2012) developed an integer programming model for airport security manpower problem and the passenger waiting time is considered with a *M/M/s* queuing model. Aickelin and Dowsland (2004) developed an indirect genetic algorithm for a nurse-scheduling problem. The difference with previous approach is that they adapt a heuristic decoder which can handle the conflict between objectives and constraints. There are few papers which deal with multi-objectives for manpower scheduling problem. For multi-objective problems, Cai and Li (2000) developed a multi-criteria optimization model with a genetic algorithm. The multiple objective functions include the minimizing of the total costs for assigning staff and maximization of surplus of staffs while satisfying the level of assigning costs, and finally minimizing the variations of staff surplus from different time periods.

From the literature review, previous manpower planning problems are commonly focuses on cost minimization approach with pre-determined possible shift time period in order to make the problem simple and easy to apply in general cases. Here, our approach do not limit the possible shifting time and therefore shifts can occur any time period unless not violating work time regulations and handling capacities of the number of passengers. To generate an optimal schedule for airport security teams by considering operating rules are difficult to achieve in a short time and time-consuming job for airport managers. The rest of the paper is organized as follows: First, Section 3 outlines an overview of the security manpower problem at airports. In Section 4, an integer programming model is developed. In Section 5, multi-objective genetic algorithm is applied to the problem. In Section 6, an airport case study is conducted with different scenario analysis. Finally, passenger monitoring simulation model is introduced and numerical results are summarized with relevant conclusions in Section 7 and 8.

3 PROBLEM DESCRIPTION

Airport security manpower scheduling primarily deal with achieving two goals: 1) Find out how many security checkpoints to open? 2) How to assign security teams in each time period? Fig. 1 illustrated a typical layout of airport security checkpoints. To generate airport security manpower schedule, numerous operational constraints are considered, including operation rules, resource capacities, and airport security levels. Especially, airport security levels have to be considered in the problem since the overall airport operation process is affected by the security levels. The screening time can be shorter or longer depends on the level of security, which changes from day to day. Moreover, an airport manager has to also consider the total costs of manpower schedule and work fairness as well. As a result, generating an optimal manpower schedule considering the above is a very difficult and time consuming task for airport managers.

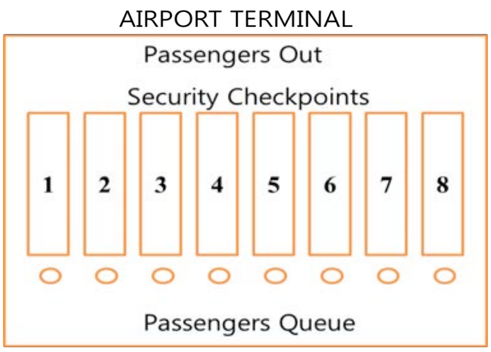


Figure 1. Airport Security Checkpoints Layout

4 AN INTEGER PROGRAMMING MODEL

In this Section, two different integer programming models are developed. Model 1 mainly focuses on cost minimization object, while Model 2 considers work fairness of security teams. The following notations are used in the models.

Sets

$i \in I$ the set of teams

$j \in J$ the set of working hours

$t \in T$ the time periods

Parameters

c_i hourly wages of team i

f_i shifting costs of team i

amax daily maximum working hours

D_t passenger demand at time period t

CAP processing capacity of a security checkpoint

Decision variables

X_{it} 1 when security team *i* shifts in time period *t*, otherwise 0.
Y_{ijt} 1 if security team *i* is assigned with *j* working hours in time period *t*, otherwise 0.
Z_{it} 1 if security team is assigned in time period *t*.
$L_{\max}$ maximum working hours of security team when considering the fairness.

The objective function of MODEL 1 is minimizing the total costs including the wages of security teams and shifting costs.

[MODEL 1] COST

$$Min \sum_{i \in I} \sum_{t \in T} (c_i \cdot Z_{it} + f_i \cdot X_{it})$$

$$\sum_{t \in T} X_{it} \leq 1 \qquad \forall i \in I \tag{1}$$

$$\sum_{j \in J} Y_{ijt} \leq X_{it} \qquad \forall i \in I,\ t \in T \tag{2}$$

$$\sum_{i \in I} Z_{it} \geq \left\lceil \frac{D_t}{CAP} \right\rceil X_{it} \qquad \forall i \in I,\ t \in T \tag{3}$$

$$\sum_{j \in J} Y_{ijt} + \sum_{k=\max\{1,t-7\}}^{t-1} (Y_{i,t-k,k} + \sum_{l=t-k+2}^{\text{amax}} Y_{ilk}) = Z_{it} \qquad \forall i \in I,\ t \in T \tag{4}$$

$$X_{it} \in \{0,\ 1\} \qquad \forall i \in I,\ t \in T \tag{5}$$

$$Y_{ijt} \in \{0,\ 1\} \qquad \forall i \in I,\ t \in T \tag{6}$$

$$Z_{it} \in \{0,\ 1\} \qquad \forall i \in I,\ t \in T \tag{7}$$

The objective function of MODEL 2 is minimizing the work time variances between security teams.

[MODEL 2] FAIRNESS

$$Min\ L_{\max}$$

$$\sum_{t \in T} X_{it} \leq 1 \qquad \forall i \in I \tag{8}$$

$$\sum_{j \in J} Y_{ijt} \leq X_{it} \qquad \forall i \in I,\ t \in T \tag{9}$$

$$\sum_{i \in I} Z_{it} \geq \left\lceil \frac{D_t}{CAP} \right\rceil X_{it} \qquad \forall i \in I,\ t \in T \tag{10}$$

$$\sum_{j \in J} Y_{ijt} + \sum_{k=\max\{1,t-7\}}^{t-1} (Y_{i,t-k,k} + \sum_{l=t-k+2}^{\text{amax}} Y_{ilk}) = Z_{it} \qquad \forall i \in I,\ t \in T \tag{11}$$

$$\sum_{t \in T} Z_{it} \leq L_{\max} \qquad \forall i \tag{12}$$

$$X_{it} \in \{0,\ 1\} \qquad \forall i \in I,\ t \in T \tag{13}$$

$$Y_{ijt} \in \{0,\ 1\} \qquad \forall i \in I,\ t \in T \tag{14}$$

$$Z_{it} \in \{0,\ 1\} \qquad \forall i \in I,\ t \in T \tag{15}$$

$$L_{\max} \geq 0 \tag{16}$$

Constraints (1), (8) ensure that one shift per day can be assigned to one security team. Constraint (2), (9) represent logical relationship between decision variables. Working hour decision variable Y_{ijt} has to occur in the same or later period than set up period X_{it}.

Constraints (3), (10) ensure that the numbers of security team has to be enough to carry the passenger demand in each time period. Constraints (4), (11) ensure the continuity of manpower scheduling which represent all possible combinations of shifting time and working hours. In this problem, each security team is assigned based on the number of passengers arriving at each time period. Thus, it is necessary to find out the set up time of security team in order to consider the continuity of scheduling while not violating the maximum working hours per day. To help understand the constraint easily, we represent an example when $t = 10$. Tab. 1 shows all possible set up time periods which are from $t = 3$ to $t = 10$. Then, we need to consider how many working hours are assigned for each period. If the set up occurs at $t = 3$, only possible option is that the security team is assigned with 8 working hours. On the other hand, when the set up occurs at $t = 10$, different working hours (1 hour to 8 hours) can be assigned since it is a start of work period in this case. Constraints (5)-(8), (13)-(15) are all decision variables and must be binary. Constraints (16) is an integer decision variable.

Table 1. Possible Shift Combinations when *t = 10*

t=10	*j (assigned working hours)*							
*t*3	8							
*t*4	7	8						
*t*5	6	7	8					
*t*6	5	6	7	8				
*t*7	4	5	6	7	8			
*t*8	3	4	5	6	7	8		
*t*9	2	3	4	5	6	7	8	
*t1*0	1	2	3	4	5	6	7	8

5 ALGORITHMS

The multi-objective manpower scheduling problem is difficult to solve and the problem has been known to be NP-complete (Cai and Li, 2000). Thus, in this paper, a multi-objective genetic algorithm (MOGA) is developed to solve multi-objective optimization problems. MOGA approaches applied to various fields where two or more conflicting objectives exist. The proposed problem deals with two objectives: minimizing the total costs and the fairness of security team's working hours. Using MOGA, it enables to show all possible set of Pareto optimal solutions to the decision makers. Thus, this paper developed the MOGA with the weighted sum approach to solve the airport security manpower scheduling problem formulated above.

5.1 Chromosome Representation

Chromosome representation is shown as Fig. 2 $T = [T_1, T_2, T_3, ..., T_{24}]$ indicates the assignment of security team in each time period. The length of chromosome denotes the possible assignments of security teams which are the maximum of 8 security teams in each time period. The required security team is determined by demand constraints (3), (9). Basically, each integer value represents team *i*, and 0 means that there is no team assignment exists. For example, chromosome T_1 only one security team is assigned by security team 5. Chromosome T_{13}, 4 security teams are assigned by team 4, 9, 8, and 11.

T_1	5	0	0	0	0	0	0	0
T_2	1	0	0	0	0	0	0	0
⋮								
T_{13}	4	9	8	11	0	0	0	0
T_{14}	1	9	8	0	0	0	0	0

Figure 2. Chromosome Representation

5.2 Fitness Function

In order to measure the performance, a fitness function is introduced. The fitness function measures the multi-objective perspectives: the total costs $f_1(x)$, and the work fairness, $f_2(x)$. Also, fitness function include penalty function $p(x)$ and it occurs when team *i* is assigned more than daily working hours (amax).

$$f_1(x) = \sum_{i \in I} \sum_{t \in T} (c_i \cdot Z_{it} + f_i \cdot X_{it}),$$

$$f_2(x) = L_{\max},$$

$$p(x) = \sum_{t}^{T} Z_{it} > \text{amax} \qquad \forall i \in I$$

Here the measure unit of two fitness functions are different. Thus, we adapt a weight-sum approach to consider multi-objective functions. Each weight factor value w_1, w_2 is selected by airport security manager. The weight assignment decision is based on the goal of scheduling between the costs and the fairness. Thus, the fitness function can be written as: $f(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) + p(x)$. Since two objectives are the minimization problem, the fitness function is transformed as $f(x) = -w_1 \cdot f_1(x) - w_2 \cdot f_2(x) - p(x)$.

5.3 Selection

A roulette wheel selection is used where each individual is assigned a portion of the wheel to its fitness. Each selected individual is included into the new population.

5.4 Crossover

The modified uniform crossover method is used to generate offspring. The crossover representation is illustrated in Fig. 3. Initially, the crossover mask is generated and randomly assigns its value as 1 or 0. If the cross mask value has 1, then offspring gene copy from the first parent value. If the crossover mask value has 0, then offspring gene copy from the second parent value. When the gene value from each chromosome has 0, then no crossover exists. The probability for performing crossover is set to 0.4.

Crossover Mask	0	0	1	0	1	0	0	0
Parent 1	1	9	16	11	0	0	0	0
Parent 2	4	9	3	0	0	0	0	0
Offspring	4	9	16	11	0	0	0	0

Figure 3. Modified Uniform Crossover

5.5 Mutation

The swap mutation method is used here which randomly selects two genes in chromosome and swaps the positions of these genes to create a new chromosome. Fig 4 represents the process of swap mutation method. When the gene value from each chromosome has 0, then no mutation exists. The probability for performing mutation is set to 0.1.

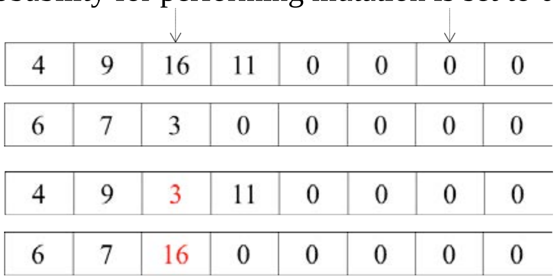


Figure 4. Swap Mutation

5.6 Feasibility Check

To ensure feasibility, it is necessary to check that the constraints (1)-(7), (8)-(15) are not violated. If not, solutions are updated as the Pareto optimal solutions. For constraints violating solutions, recursive operation is performed until that becomes feasible solutions.

The procedure of the MOGA is summarized as follows.

- Step 1. Generate an initial population containing N_{pop} string where N_{pop} is the number of strings in each population.
- Step 2. Calculate the values of the objective functions for the generated strings. Update a tentative set of Pareto optimal solutions.
- Step 3. For selection, we used a weighted sum approach which combines multiple objective functions into a scalar fitness function. Thus, the weights of objective functions are randomly assigned. Thus, $f(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) + p(x)$.
- Step 4. Since two objectives are minimization problem, the equation is transformed as $f(x) = -w_1 \cdot f_1(x) - w_2 \cdot f_2(x) - p(x)$.
- Step 5. For selection, select a pair of individuals form the current population using the roulette wheel selection strategy, which is a method to reproduce a new generation proportional to the fitness of each individual.
- Step 6. For crossover, randomly selected uniform crossover is applied in this case. When the gene value has 0, then no crossover exists.
- Step 7. For mutation, a swap mutation is selected in this case. When the gene value has 0, then no mutation exists.
- Step 8. The elite strategy retains the top k solutions in order to keep quality solutions in each generation and updates the Pareto optimal solutions.
- Step 9. If the terminating condition is satisfied, then stop, otherwise go to step 2.

5.7 Parameter Setting

For the MOGA experiments, the following parameters are used, see Tab. 2.

Table 2. MOGA Parameter Setting

Population Size	100
Generations	200
Crossover Rate	0.4
Mutation Rate	0.1
Elitism	1

6 AIRPORT CASE STUDY

To validate our methods, we obtained real data from one of the largest airport in Northeast Asia. A week period of passenger arrivals data was collected and used it as an input data for passenger demand. Fig. 5 illustrated the number of passengers in each time period and shows two main peak time periods in the morning and evening. Several scenario tests have done by considering multiple airport security levels and security team's possible working hour allowances.

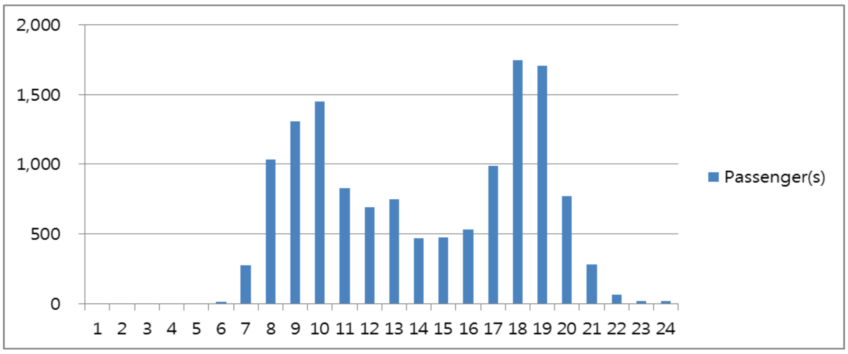


Figure 5. Number of Passengers

6.1 Airport Resources and Operational Rules

The process of airport security checkpoints is separated each other independently, therefore the operation is timely differentiated based on the number of passengers arriving. Thus, the operations with full resources are needed in a peak time, while the minimum levels of resources are recommended in non-peak time period. Airport security operation is commonly achieved by team-based, which includes different duties of workers. Typical duties include managing overall processes, reading X-rays, responsible for hand detect, and helping with passengers' carry-on items. The working time regulations set a maximum limit on daily working hours by regulation. Work hours is limited as 8 hours per day, however it can be extended up to 10 hours per day with an employee's agreement. Therefore, this paper also considers the situation when work hours allowance changes from 8 hours to 10 hours.

6.2 Airport Security Levels

Airport security levels changes day to day and the relevant procedures vary. In our case, airport security levels divided into five different levels: Green, Blue, Yellow, Orange, Red (http://english.mltm.go.kr/intro.do). Based on the security levels, required steps are different and passenger screening time gets longer or shorter. After discussions with an airport manager, we determined the expected screening time for each security level. For Green, Blue, Yellow level, we assume that the screening time for each passenger is between 15 seconds to 20 seconds. For Orange, the screening time increased between 20 seconds to 30 seconds. Finally for red, the screening time increased to 30 seconds to 50 seconds. Airport security checkpoints are separated each other and its process occurs independently. The operation of security lines are timely different based on the number of passengers. Thus, operations with full resources are needed in a peak time, while the minimum levels of resources are recommended in non-peak time period. In this paper, we consider 1 passenger terminal with 8 security checkpoints.

6.3 Test Environments

The proposed integer programming model was coded in a C language program with Visual Studio 6.0. The computational experiments were performed with ILOG CPLEX v. 11.2 on a 2.4 GHz Intel Core i5 CPU PC with 4 GB RAM. The following parameters are used as input value. Set up costs: $1000, Hourly wages: $106/hr, Number of Security Checkpoints: 8. Fig. 7. shows the output of the experiments for Model 1 and Model 2.

6.4 Test Results

Fig. 6 shows the results from the integer programming model. Each security team is assigned for certain time periods to fulfil the number of passengers arriving. Model 1, which is the cost minimization approach tend to reduce the set up numbers to reduce the total operating costs. On the other hand, fairness model consider the work fairness between each team. As it shown, the number of working hours is more balanced compare to the cost minimization model. The computation times for Model 1 and Model 2 are 0.14 seconds and 7.36 seconds, respectively, while the computation time of the exact algorithm (enumeration algorithm) is almost 7 hours.

TIME PERIOD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	TOTAL ASSIGNED
TEAM	11	11	11																						3 hrs
				3	3	3	3	3	3	3	3														8 hrs
							12	12	12	12	12														5 hrs
								13	13	13	13	13	13												6 hrs
								9	9	9	9	9	9												6 hrs
								6	6	6															3 hrs
								16	16	16	16	16	16	16											7 hrs
									2	2															2 hrs
													5	5	5	5	5	5	5	5					8 hrs
														15	15	15	15	15	15	15					7 hrs
															10	10	10	10	10						5 hrs
																	14	14	14						3 hrs
																		4	4						2 hrs
																		7	7	7	7	7	7	7	7 hrs
																		8	8						2 hrs
																			1	1					2 hrs

TIME PERIOD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	TOTAL ASSIGNED
TEAM	10	10	10	10	10																				5 hrs
						5	5	5	5	5	5														6 hrs
							12	12	12	12															4 hrs
								1	1	1	1														4 hrs
								13	13	13	13	13	13												6 hrs
								14	14	14	14	14	14												6 hrs
								6	6	6	6	6	6												6 hrs
									11	11	11	11	11	11											6 hrs
													3	3	3	3	3	3							3 hrs
													9	9	9	9	9	9							6 hrs
														2	2	2	2	2							5 hrs
														16	16	16	16	16	16						6 hrs
															7	7	7	7	7						5 hrs
																8	8	8	8						4 hrs
																	15	15							2 hrs
																		4	4	4	4	4	4		6 hrs

Figure 6. Airport Security Teams Schedule (MODEL 1, MODEL2)

Fig. 7 represents the results of experiments with different security levels and team’s working hour allowances based on MODEL 1. From the results, the total costs of airport security check process goes up when the level of security increases above the Yellow level. Possible savings on the total operating costs can be achieved by increasing the daily working hour’s allowances from 8 hours to 10 hours.

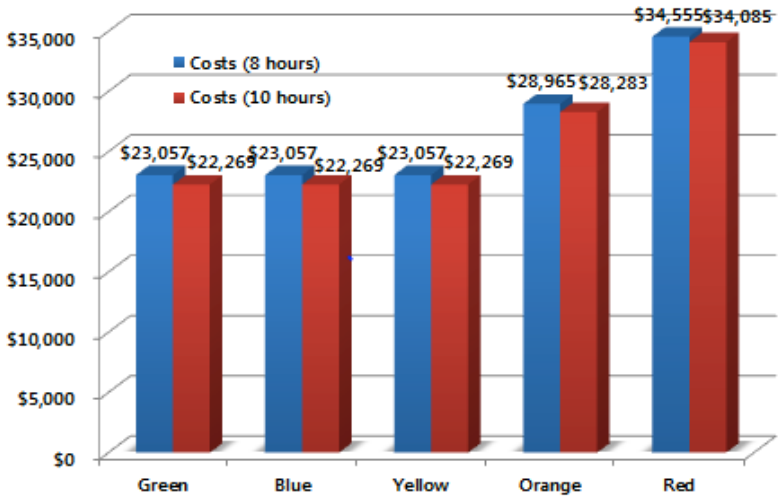


Figure 7. Scenario Test Result

7 AIRPORT PASSENGERS MONITORING

7.1 Passenger Imbalance Problem

In this Section, to present the implementation of our manpower scheduling method, we consider the situation when the numbers of arriving passengers are imbalanced from terminal to terminal and conduct the simulation experiments with real data. The passenger imbalance issue is quite common in most of airports due to different airlines dominate certain terminals. Therefore, we take the idea of passenger monitoring system which provide the estimated time for each terminal. Fig. 8 shows the concept of passenger monitoring at airports, which provides passengers an estimated security screening process time for each terminal. The advantage of the passenger monitoring method is that passenger can move to terminals where less waiting time is expected. As a result, an overall passenger waiting time can be balanced between terminals and the total passenger waiting time at airports can be reduced.

Figure 8. Passenger Monitoring Service

Based on the idea the above, we developed a simulation model to test an efficiency of the passenger monitoring. For the current airport, a security check takes a minimum of 10.68

seconds and a maximum of 1.98 minutes per passenger. The main reason for the variation is the gates occupied with different airlines and peak time periods. We conducted a survey among 211 passengers at the airports in order to examine passenger behavior. Fig. 9 represents the output of passengers responded when faces with a choice for changing the gate to save time. The results show that 69% of passengers are willing to move to the nearest departure gates based on information provided by the passenger monitoring service. The simulation model considers the survey results as a dispatching rule.

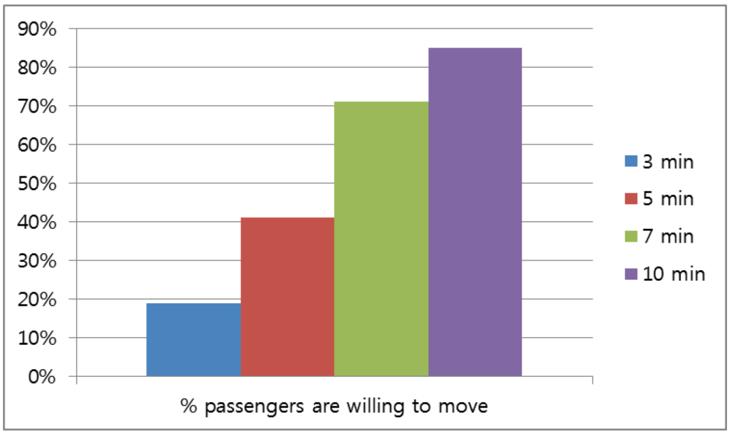


Figure 9. Passenger Survey Results

To run the simulation model the following values are assumed, in Tab 3.

Table 3. Simulation Setting

Arrival rates	Non-stationary Poisson processes
Security check	[15, 22] time (sec.)
Departure Terminals	4 terminals
Security checkpoints	32 security checkpoints

7.2 Test Results

Based on the above assumptions, the simulation model runs 30 times. The average time spent in the security check process with the passenger monitoring system is given in Fig. 10. Compared to the data from the case study the total time spent in security check process is reduced from, on average 8.684 minutes to 8.271 minutes per passenger, which is the equivalent of a 41.33% reduction. The reductions for the other gates are as follows: Departure Gate B = 0.237 minutes per passenger, 23.71%, Departure Gate C = 0.179 minutes per passenger, 17.85%, and Departure Gate D = 0.363 minutes per passenger, 36.26% respectively.

Case	Gate	Service Time	Wait Time	Total Time
Current Situation	A	0.418	8.267	8.684
	B	0.419	4.743	5.162
	C	0.418	3.570	3.987
	D	0.4.70	7.253	7.670
	Average	0.418	5.958	6.376
Monitoring System	Gate	Service Time	Wait Time	Total Time
	A	0.418	7.854	8.271
	B	0.419	4.505	4.925
	C	0.418	3.391	3.809
	D	0.417	6.890	7.307
	Average	0.418	5.660	6.078

Figure 10. Simulation Results (in minutes)

In worst case scenario, the total screening time on average gates reduced from 30.445 minutes to 28.922 minutes per passenger. Thus, the proposed monitoring method helps to improve security line balance problem in peak time period.

8 CONCLUSION

In this paper, we developed an integer programing model and applied multi-objectives genetic algorithms to solve manpower scheduling problem arising at airports. We applied the developed methods to real airport operation using real data obtained from the airports. The computational results showed that the proposed approach solve the problem in a short time and promising near optimal solutions. For the extension, this paper suggests methods to relieve passenger imbalance problem at airports by introducing passenger monitoring methods based on simulations. Our methods can be implemented as a decision making tool for airport security operators to test current status of airport security conditions and to find out the better manpower schedule in a short time. For future studies, it would be interesting to apply to other meta-heuristic, such as simulated annealing, and tabu search and compare the performances. Finally, the proposed research can be also applied to the future airport security system where each process differs based on the group of passengers.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2009-0076365)

REFERENCES

Holloran T., Byrn J. (1986), United Airlines Station Manpower Planning System, *Interfaces*, **16**, 39-50.

Schindler S., Semmel T. (1993), Station Staffing at Pan American World Airways, *Interfaces*, **23**, 91-98.

Brusco M., Jacobs L. W., Bongiorno J. R., Lyons V. D., and Tang B. (1995), Improving Personnel Scheduling at Airline Stations, *Operating Research*, **43**, 741-751.

Cai L., Li K. N. (2000), A Genetic Algorithm for Scheduling Staff of Mixed Skills under Multi-Criteria, *European Journal of Operational Research*, 125, 359-369.

Yang T., Yan S., Chen H. (2003), An Airline Maintenance Manpower Planning Model with Flexible Strategies, *Journal of Air Transport Management*, **9**, 233-239.

Aickelin U., Dowsland K. (2004), An Indirect Genetic Algorithm for a Nurse-Scheduling Problem, *Computer and Operations Research*, 31, 761-778.

Huang H. C., Lee C. and Xu Z. (2006), The Workload Balancing Problem at Aircargo Terminals, *OR Spectrum*, **228**, 705-727.

Ernst A. T., Jiang H., Krishnamoorthy M., and Sier D. (2004), Staff Scheduling and Rostering: A review of Applications, Methods, and Models, *European Journal of Operation Research*, **153**, 3-27.

Lee L. H., Lee C. U., Tan Y. P. (2007), A Multi-Objective Genetic Algorithm for Robust Flight Scheduling using Simulation, *European Journal of Operation Research*, **177**, 1948-1968.

Rong A., Grunow M. (2009), Shift Designs for Freight Handling Personnel at Air Cargo Terminals, *Transportation Research Part E.*, **45**, 725-739.

Stolletz R. (2010), Operational workforce Planning for Check-in Counters at Airports, *Transportation Research Part E.*, **46**, 414-425.

Seo S., Choi S., and Lee C. (2012), Security Manpower Scheduling for Smart Airports, *Lecture Notes in Electrical Engineering*, **114**, 519-527.

A MULTI-OBJECTIVE MODEL AND A MEAT-HEURISTIC FOR REDUCING TRUCK EMISSIONS AT CONTAINER TERMINALS

Gang Chen[1], Kannan Govindan[2] and Mihalis M. Golias[3]

[1]*Department of Environmental and Business Economics, University of Southern Denmark Niels Bohrs Vej 9, Esbjerg, Denmark, gch@sam.sdu.dk*

[2]*Department of Business and Economics, University of Southern Denmark, Campusvej 55, Odense, Denmark, gov@sam.sdu.dk*

[3]*Department of Civil Engineering, University of Memphis, 112A Engineering Science Bldg, 3815 Central Avenue, Memphis, the United States, mihalisdgolias@gmail.com*

ABSTRACT

A seaport is a significant source of exhaust emissions and air pollution for the local region. This study addresses the truck emissions in container terminals and investigates the trade-off between truck emissions and truck arrival patterns. First, a queueing network based on fluid-based approximations is proposed to analyze time-dependent truck queueing processes at the gate and the yard of a container terminal. Then, a multi-objective optimization model is developed to minimize both truck waiting times and number of shifted arrivals. Furthermore, emissions reduction from truck engine idling is analyzed through a post-optimization phase of truck waiting time. Lastly, a real case study demonstrates that truck emissions could be significantly reduced by a small shift in truck arrivals.

Keywords: container terminal, truck emissions, multi-objective optimization, queueing network

1 INTRODUCTION

While making a considerable contribution to the economic growth, a seaport is a major source of emissions in the local region. The emissions include air pollutants such as nitrogen oxide (NO_X), sulphur oxide (SO_X) and particulate matter (PM), which degrade the atmospheric environment in the vicinity of port areas and have a direct effect on the human population living in nearby cities and coastal communities[1]. For example, Saxe and Larsen (2004) investigate air pollution in three Danish seaports and conclude that emitted pollutants could induce health problems to people living or working near ports. Other studies, with similar results on port-related emissions, have been conducted at other seaports, e.g. Port of Los Angeles (Starcrest Consulting Group, 2011), Port of Piraeus (Tzannatos, 2010), and the Belgian ports (Meyer *et al.*, 2008).

At a seaport, the sources of emissions include ocean-going vessels, harbor crafts, cargo handling equipment, rail locomotives and heavy-duty vehicles. Comparatively, truck emissions have probably have more harmful impact – relative to the emitted amounts – when emitted near local communities (Saxe and Larsen, 2004). There are a number of studies and industry reports on truck emissions reduction efforts at seaports. Some of these studies focus on the impact of engine technologies on emission factors, e.g. Truck Stop Electrification (Zietsman *et al.*, 2009); while the other studies focus on different mechanisms to coordinate

[1] http://www.maine.gov/dep/blwq/topic/vessel/airemissionsreport.pdf

terminal and truck operations, e.g. Terminal Appointment System (TAS), tariff/toll pricing policy, and Vessel-Dependent Time Windows (VDTWs) for truck entries. These economic or operational mechanisms shift truck arrivals from peak times to off-peak times, in order to achieve an optimal arrival pattern. How to identify the optimal arrival patterns becomes the common backbone of these mechanisms. This optimization involves the trade-off between the emissions due to a truck arrival pattern and the number of shifted truck arrivals this pattern requires. Such trade-off relationship will probably be terminal-specific, as different terminals have different operational policies while truckers have different preferences and relationships with each terminal (even at the same port).

This paper presents a methodology for investigating the relationship between truck arrival shifts and emissions reduction at a marine container terminal. First, a queueing network based on fluid approximations is proposed to model the time-dependent truck queueing processes at the gate and the yard of the terminal. Then, a multi-objective optimization model is developed to minimize both truck turn times and the number of shifted truck arrivals. Furthermore, emissions reduction from truck engine idling is analyzed through a post-optimization phase. In order to solve the multi-objective optimization model, we develop a GA based heuristic. Lastly, a number of computational examples are performed to critically discuss the proposed methodology.

2 LITERATURE REVIEW

There is a rich literature on reducing truck emissions at seaport areas. In this section we focus on studies related to truck turn time estimation and truck arrival pattern optimization. Truck turn time is defined as the duration from the arrival of a truck at the terminal gate to the moment of exit (i.e. the summation of truck waiting time at the gate and truck service (un/load container) and waiting time at the yard). Existing studies identify three approaches to estimate truck turn time and its components: a) simulation based (Huynh *et al.*, 2004; Huynh, 2009), b) regression models (Huynh *et al.*, 2004; Goodchild and Mohan, 2008), and c) queueing models (Guan and Liu, 2009; Chen *et al.*, 2011a; Chen *et al.*, 2011b).

Huynh *et al.* (2004) developed a discrete event simulation model of a container terminal, representing the precise movements of trucks and yard cranes. The model was used to find the number of yard cranes needed to achieve a desired truck turn time. Huynh (2009) developed a simulation model to evaluate performance of various rules for truck arrival management. Both models are powerful tools to present detailed operations of trucks and terminal equipment and can be used to test different operation scenarios. But long computational time and a large number of replications are the major barrier for integrating a simulation model into an optimization process.

Huynh *et al*. (2004) applied regression analysis to the output of their simulation model for different scenarios of container terminal operations. Goodchild and Mohan (2008) discussed the predictive accuracy of the model by Huynh *et al*. (2004) and claimed that using averages rather than a single simulation replication reduces the variability and improves the fit of the regression model (i.e. high R2 value). Both studies lead to the conclusion that statistical based models are not accurate enough to estimate truck turn times at a container terminals.

There are two types of queueing models used to estimate truck turn times: a) conventional stationary queueing models, and b) non-stationary queueing models. Guan and Liu (2009) analyzed truck queues at marine container terminal gates with a stationary M/Ek/c queueing model. A limitation of stationary queueing models is they neglect the transient behavior and can analyze only the steady state of a queue. This raises concerns about the applicability of simple stationary models (Green and Kolesar, 1991). Typically, truck queues at marine container terminal gates are not in steady state, as truck arrival and gate service rates vary over time. To capture the time-varying behavior of such queueing processes, state-dependent queueing models are effective and robust tools (Smith, 2010). In a pioneering study by Green and Kolesar (1991), a point-wise stationary approximation (PSA) was proposed to model non-stationary queueing systems. Whitt (1991) further verified that the PSA model is asymptotically correct as the service and arrival rates increase with fixed instantaneous traffic intensity. As a further development of PSA, a point-wise stationary fluid flow approximation (PSFFA) was proposed by Wang et al. (1996) by combininging PSA and a fluid flow model, to analyze single server non-stationary queueing models. In this study, PSFFA was proved to have better accuracy than PSA. PSFFA relies on invertible steady state functions and therefore is not capable to analyze most multi-server non-stationary queueing models. To address the issue of inverting complex queueing functions, Chen *et al*. (2011a) propose integrating the bisection method into PSFFA for the multi-server non-stationary M(t)/Ek(t)/c(t) queue. This approximation, labeled as B-PSFFA, can reach a high accuracy. The above models consider only truck queues at the terminal gates, but ignore queues at the yard. To address both types of truck queues, another group of researchers (Chen *et al*., 2011b) developed a two-layer queueing network. This queueing network can model truck queues at the gate and the yard jointly. However, there are two limitations of this queueing network: first, the assumption that the gate service times follow an Exponential distribution is not based on any empirical analysis; and second, the assumption that a gate system comprises of multiple independent queueing systems does not comply with the practice where the truck queue is served simultaneously by all the gate lanes. So ideally a gate system should be modeled as a multi-server queueing system.

The second part of the relevant literature focuses on truck arrival pattern optimization studies. Huynh and Walton (2008) integrated a search heuristic with the simulation model developed by Huynh *et al*. (2004) to reduce average truck turn times to a target level specified by the terminal operator. This optimization model has the disadvantage of long computational times due to an embedded simulation model. Chen *et al*. (2011b) developed a queueing network to model the truck queues at the gate and the yard jointly. Furthermore, they integrated the queueing network into an optimization model. The first component of their objective function may be difficult to calculate as, in practice, original truck arrival times are not known during the planning phase. Nevertheless, to the best of our knowledge, Chen *et al*. (2011b) is the only existing study modeling and optimizing truck flows at a marine terminal using a non-stationary queueing network. Another two studies proposed optimization models only for the truck flows at the gates. The first one, Guan and Liu (2009) developed an optimization model using a stationary queueing model, which results in an evenly distributed pattern of truck arrivals. This over-adjusted result requires too many trucks to change schedule The second, Chen et al. (2011a) improved Guan and Liu’s model (2009) with the non-stationary M(t)/Ek(t)/c(t) queue. The result of the model is a time-dependent pattern of truck appointments, which requires only a small number of trucks to change schedule. The above optimization models are all single objective. In practice, truck arrival optimization often involves the trade-off between contradicting objectives, e.g. number of shifted arrivals and truck queue length. Understanding the trade-off will help the terminal operator to better manage truck arrivals.

From the above literature review, a gap can be seen, where none of the existing models/methods is able to accurately estimate truck turn times in a computationally efficient fashion. In addition, the optimization models found in the current literature are single objective and do not consider trade-offs between emissions reduction and shifted truck arrivals. In this study, we try to fill this gap.

3 PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

3.1 Problem Description

This problem description and the mathematical model have been developed in an early version of this work (Chen and Govindan, 2011). In our problem, the container terminal has one entrance gate with multiple lanes. Typically, a truck arrives at a terminal gate during its preferred or assigned appointment time window, and the arrival rate varies from hour to hour. The gate serves trucks on the 'first come, first served' principle. The gate system can be treated as a multi-server non-stationary $M(t)/E_k(t)/c(t)$ queueing system as it has been assessed that truck inter-arrival times typically follow an exponential distribution and that gate service times follow an Erlang distribution (Guan and Liu, 2009). Although truck inter-arrivals may follow some different distributions at different terminals (Huynh *et al.*, 2011), state-dependent queueing models are insensitive to the input distribution (Smith, 2010). After finishing the entry paperwork, a truck proceeds to a designated yard zone and waits in a queue, if any, until an available yard crane or straddle carrier come to (un)load its container. Finally, it departs at an exit gate. The yard service operation follows a non-Poisson process and should be represented as an M/G/1 queue (Kim and Kim, 2002). Therefore, the yard system is treated as a combination of multiple independent non-stationary $M(t)/G(t)/1$ queues. Since truck queues occur very often at both the gate and the yard generating air pollutants and GHG emissions, it is necessary to maintain queue lengths under a certain level by shifting some truck arrivals from peak to off-peak periods.

To model the truck queueing processes at the gate and the yard, a queueing network is developed combining the above two queueing systems, as shown in Figure 1. The gate system is modeled as a multi-server queueing system, and the gate service process is modeled as an Erlang distribution with parameters obtained from empirical analysis.

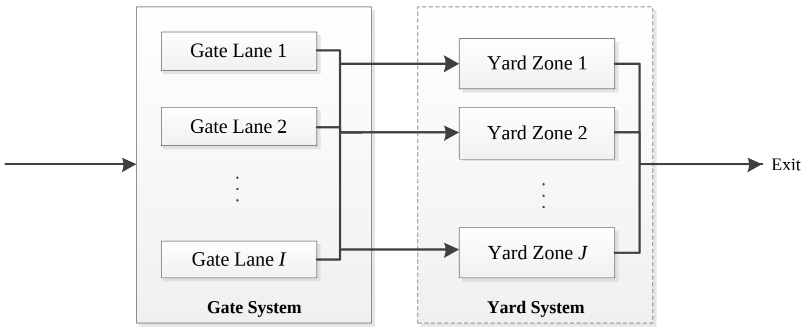


Figure 1. Queueing network in a container terminal

The proposed queueing network is solved with a fluid-based approximation model, called point-wise stationary approximation (Green and Kolesar, 1991; Wang *et al.*, 1996), which has received increasing attention in the last decade for modeling non-stationary queueing systems in the fields of electrical engineering and management science. What 'point-wise' refers to is that, a time period is divided into smaller time intervals and then stationary approximations are used to measure the system performance at each time interval. This fluid based approximation model is comprised of three major components: a flow balance function, an exit flow function, and a time dependent capacity utilization ratio function as shown in Equation 1, 2 and 3 respectively. Equation 1 states that during a state transition the change of queue length is equal to arrivals minus departures. Equation 2 calculates the actual discharge rate at a time point based on the estimated capacity utilization ratio. Equation 3 estimates the capacity utilization ratio at the time point based on the related queue length, where $g^{-1}(.)$ is the inverse function of the correspondent stationary queueing model.

$$l_t = l_{t-1} + \lambda_{t-1} - d_{t-1} \tag{1}$$

$$d_t = min\left(l_t,\ c_p \rho_t \mu_t\right) \tag{2}$$

$$\rho_t = g^{-1}(l_t) \tag{3}$$

Regarding shifting truck arrivals, a potential conflict may exist between the reduction of truck emissions and the convenience of truckers. To investigate the trade-off between emissions reduction and shifted truck arrivals, we propose a two-phase optimization approach. Phase I is used to find all the system-optimal truck arrival patterns, and Phase II is used to estimate the truck emissions reduction based on saved truck idling time for every obtained optimal pattern. All the input data, derived variables and decision variables are as below.

Input parameters		
	p	index of time periods, $p = 1, \dots, P$, where P is the number of periods;
	t	index of fluid-based modeling time point $t = 1, \dots, T$;
	i	index of gate lanes, $i = 1, \dots, I$, where I is the number of gate lanes;
	j	index of yard zones, $j = 1, \dots, J$, where J is the number of yard zones;
	c_p	number of gate lanes in service in time period p, $c_p \leq I$;
	λ_p^g	number of preferred truck arrivals in time period p;
	μ_{it}^g	processing rate of gate lane i at time point t (trucks);
	μ_{jt}^y	processing rate of yard zone j at time point t (trucks);
	k	Erlang distribution parameter for gate service time;
	v	coefficient of variation of yard service time distribution;
	e	emission factor of truck engine idling (g/hour);
	β_j	yard destination proportion;
Derived variables		
	λ_t^g	number of truck arrivals at gate at time point t;
	ρ_{it}^g	utilization rate of gate lane i at time point t;
	l_t^g	queue length of trucks at gate at time point t;
	w_t^g	waiting time of trucks at gate at time point t;
	d_{it}^g	discharge rate of gate lane i at time point t;
	a_t^g	traffic intensity of the gate system at time point t;
	λ_{jt}^y	number of truck arrivals at yard zone j at time point t;
	ρ_{jt}^y	utilization rate of yard zone j at time point t;
	l_{jt}^y	queue length of trucks at yard zone j at time point t;
	w_{jt}^y	waiting time of trucks at yard zone j at time point t;
	d_{jt}^y	discharge rate of yard zone j at time point t;
Decision variable		
	$\overline{\lambda_p^g}$	truck arrivals allowed in time period p.

3.2 Phase I: Truck Arrival Pattern Optimization

There are three objectives when managing truck arrivals: (1) minimizing the total number of shifted arrivals to reduce inconvenience to the truckers; (2) minimizing the total truck waiting time to reduce truck emissions[2], and (3) minimizing the maximum truck waiting time[3]. We develop an optimization model with the above three objectives, as shown in Equations 4 through 6. Note that objective Z_2 in Equation 5 is measured as the sum of queue lengths multiplied by unit time period, which is equivalent to the total truck waiting time.

$$\min Z_1 = \sum_t \left|\overline{\lambda_p^g} - \lambda_p^g\right|/2 \tag{4}$$

$$\min Z_2 = \sum_t l_t^g + \sum_t \sum_j l_{jt}^y \tag{5}$$

$$\min Z_3 = max\left(w_t^g\right) + max\left(w_{jt}^y\right) \tag{6}$$

Arrival Pattern Constraints

Equation 7 ensures that total truck volumes after optimization equals the original demand. Equation 8 ensures the number of appointment quotas in each time period is nonnegative. Equation (9) calculates the number of trucks arriving at gate at time point *t*.

$$\sum_p \overline{\lambda_p^g} = \sum_p \lambda_p^g \tag{7}$$

$$\overline{\lambda_p^g} \geq 0 \tag{8}$$

$$\lambda_t^g = \overline{\lambda_p^g}\ (P/T)^{-1} \tag{9}$$

Constraints at the Gate

Equation 10 calculates the waiting time of trucks arriving at time point t based on the queue length and the discharge rates of the lanes; Equation 11 ensures that during a state transition the change of queue length is equal to arrivals minus departures. Equation 12 calculates truck discharge at gate lane i at time point t. The utilization ratio of the gate lane i at time point t is calculated in Equation 13. Equation 14 approximates the traffic intensity (a_t^g) using the inverse function of the well-known Cosmetatos' formula (Cosmetatos, 1976) and a correction factor γ_t based on the queue length l_t^g. This approximation is a crucial part of the model, and will be described in details in the next subsection.

$$w_t^g \approx l_t^g \Big/ \sum_i d_{it}^g \tag{10}$$

$$l_{t+1}^g = \max\left(l_t^g + \lambda_t^g - \sum_i d_{it}^g, 0\right) \tag{11}$$

$$d_{it}^g = \mu_{it}^g \times \rho_{it}^g \tag{12}$$

$$\rho_{it}^g = a_t^g / c_p \tag{13}$$

$$a_t^g = \text{Cosmetatos}^{-1}\left(l_t^g, c_p, k\right) \times \gamma_t \tag{14}$$

B-PSFFA Approximation

If truck arrivals follow a Poisson distribution and gate service times have an Erlang distribution, the gate system can be analyzed using Cosmetatos' formula. As shown in Equation 15, Cosmetatos' formula is too complex to invert.. Therefore, the Bisection method and the correction factor γ_t (see Equation 16are used as an alternative to inverting the complex formula, as proposed by Chen *et al.* (2011a). Their method, called B-PSFFA, is an accurate and fast approximation for non-stationary $M(t)/E_k(t)/c(t)$ queues. In B-PSFFA, the

[2] Truckers usually keep their engines idling while waiting in a queue
[3] Maximizes truckers opportunities for multiple loads a day

traffic intensity a_t^g is estimated with the bisection method, which repeatedly tries the midpoint of its value range. Given a midpoint of the value range $\overline{a_t^g}$, one can calculate the corresponding queue length using Equation 16 and Equation 15. Comparing this calculated queue length with the 'real' queue length l_t^g, one can easily identify whether the real traffic intensity a_t^g lies on the left side of the midpoint $\overline{a_t^g}$ (or on the right side), and then delete the right half range (or the left half range). Repeating this operation for a number of times, one can get an accurate enough estimation of the traffic intensity a_t^g.6.

$$l_t^g = \frac{\left(a_t^g\right)^{c_p+1}}{\left(c_p-1\right)!\left(c_p-a_t^g\right)^2}\left\{\sum_{n=0}^{c_p-1}\frac{\left(a_t^g\right)^n}{n!}+\frac{\left(a_t^g\right)^{c_p}}{\left(c_p-1\right)!\left(c_p-a_t^g\right)}\right\}^{-1}\left\{\frac{1+\frac{1}{k}}{2} + \frac{\left(1-\frac{1}{k}\right)\left(1-\frac{a_t^g}{c_p}\right)\left(c_p-1\right)\sqrt{4+5c_p}-2}{32a_{tj}}\right\} \quad (15)$$

$$a_t^g = \overline{a_t^g}\times\gamma_t = \overline{a_t^g}\times\left(1-1.09\times\left(c_p{}^{0.866}\times\left(\sum_i \mu_{it}^g\right)^{1.045}\right)^{-1}\right) \quad (16)$$

Constraints at the Yard (PSFFA Approximation)

The number of trucks arriving at yard zone *j* at time point *t* is calculated based on the gate lane discharge rates and a predefined (time-invariant) yard destination proportion shown in Equation 17. Equation 18 states that during a state transition the change of queue length is equal to arrivals minus departures. Equation 19 calculates the departures from yard zone *j* at time point t. In Equation 20 the utilization ratio of the yard zone *j* at time point t is computed by the PSFFA approximation proposed by Wang *et al.* (1996). Equation 21 estimates the truck waiting time at yard zone *j* at time point t.

$$\lambda_{jt}^y = \sum_i d_{it}^g\times\beta_j \quad (17)$$

$$l_{j(t+1)}^y = \max\left(l_{jt}^y+\lambda_{jt}^y-d_{jt}^y, 0\right) \quad (18)$$

$$d_{jt}^y = \mu_{jt}^y\times\rho_{jt}^y \quad (19)$$

$$\rho_{jt}^y = \left(l_{jt}^y+1-\sqrt{\left(l_{jt}^y\right)^2+2\times v^2\times l_{jt}^y+1}\right)/\left(1-v^2\right) \quad (20)$$

$$w_{jt}^y \approx l_{jt}^y / d_{jt}^y \quad (21)$$

3.3 Phase II: Truck Emission Estimation

The above truck arrival coordination can reduce the total truck waiting time (the second objective in Phase I), hence emissions from truck idling can also be reduced. Note that this study focuses on truck idling emissions, and not emissions produced while trucks are moving. The latter is depend on the terminal layout and cannot be reduced significantly through truck arrival coordination. The reduced truck idling emissions can be estimated by Equation 22, where *e* is the matrix of engine idling emission factors. These emission factors, shown in Table 2, were obtained from POLA (Starcrest Consulting Group, 2011).

$$\text{Emissions reduction} = (\text{original total waiting time} - \mathrm{Z}_2)\times e \quad (22)$$

Table 2. Emission Factors of Truck Engine Idling (g/hour)

CO_2	N_2O	CH_4	PM_{10}	$PM_{2.5}$	DPM	NO_X	SO_X	CO	HC
4,640	0.037	0.18	0.22	0.2	0.2	94.8	0.04	16.82	6.24

4 SOLUTIONS ALGORITHM

To solve the above model, a GA based heuristic (GABH) is proposed in this paper, consisting of four parts: a) the chromosomal representation; b) the chromosomal mutation; c) the chromosomal selection process, and d) the stopping criterion. In this paper, we use an integer chromosomal representation, and an illustration of the chromosome structure is given in figure 2. As seen in figure 2, the chromosome has ten cells; one for each hour that the terminal gates are open. Each cell contains an integer number that represents the number of trucks that arrive within that time period.

4.1 Chromosomal Mutation

In the proposed GABH we adopt 6 types of mutation referred to as swap, insert, spread, combine, greedy and mean mutation. The first four mutations are depicted in figure 2 for an example of 10 time periods. These four mutations are applied to every chromosome generating four mutated children per parent chromosome. The last two mutations are only applied to a selected number of chromosomes as they try to address the issue of discontinuity in the Pareto Front (PF). To select the chromosomes for the latter two mutations we calculate a Euclidean distance based criterion between consecutive chromosomes (in terms of the value of the first or second objective function). The goal is to have a PF without large gaps between two consecutive solutions. We next describe the distance criterion and the latter two mutations (greedy and mean).

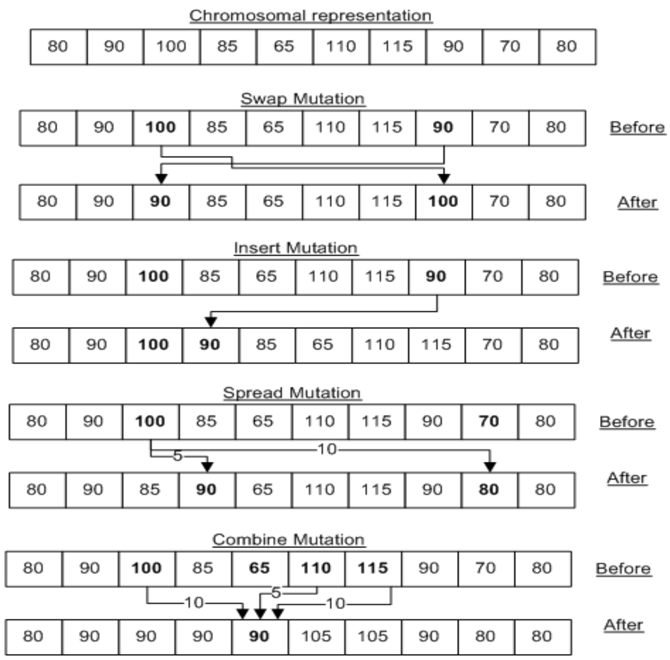


Figure 2. Chromosome representation and basic mutation procedures

Euclidean distance based criterion

Let *chrom*={x_1,x_2,..............x_j} be a set of chromosomes that belong to the PF during an iteration of the GABH, sorted in increasing order of the first (or second) objective function.

Let $f_i^n(x_j) = \dfrac{f_i\ (x_j)}{\max\limits_{j' \in chrom}(f_i\ (x_{j'}))}\ \forall i \in \{1,2,3\},\ j \in chrom$ be the normalized value of objective function i for chromosome x_j. We define: $D(x_j, x_{j+1}) = \sqrt{\sum_{i \in \{1,2,3\}} \left(f_i^n(x_j) - f_1^n(x_{i+1})\right)^2},\ i \in chrom$ as the distance between chromosome j and j+1. If $D(x_j, x_{j+1}) > \dfrac{\sum_j D(x_j, x_{j+1})}{|chrom|}$ where $|chrom|$ is the cardinality of the set *chrom* (i.e. the number of chromosomes at the current iteration) we perform the two additional types of mutation (greedy and mean) to chromosomes x_j and x_{j+1}. These two mutations are as follows.

Greedy mutation

The steps of the greedy mutation are showcased in figure 3 with an example of 10 time periods. The third and sixth period have been randomly selected as part of the first step of the greedy mutation (i.e. k=3 and k'=6). In this case $p=|100\text{-}110|=10$ and $a=U(1,10)=9$. During steps 3 through 7 parts of demand a (i.e. 9 trucks) is randomly allocated to the period with the smallest demand. First, 6 trucks (β=U(1,9)=6) are moved to time period 5 (which was the smallest-demand time period and now increased from 65 to 71) and then 3 trucks (β=U(1,3)=3) are moved to time period 9 (which was the new smallest-demand time period and now increased from 70 to 73). The demand in time period 6 decreased from 110 to 101 as 9 trucks were moved to the other two time periods.

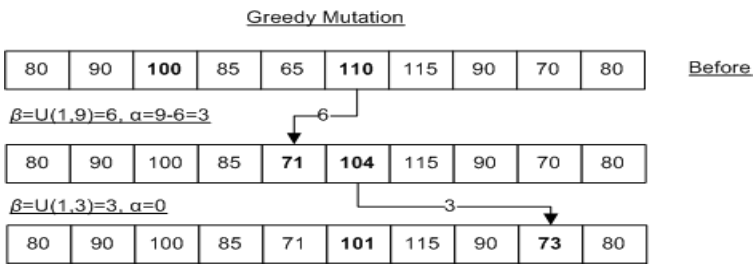


Figure 3. Greedy mutation example

Mean mutation

The four steps of the mean mutation are showcased in figure 4 for the same example as in figure 3. The sixth and seventh period have been randomly selected as part of the first step (i.e. k=6 and k'=7). Their demand difference is to be split equally between them (in step 2). In this example as the difference in the demand is an odd number (condition of step 3 is true) we randomly select the sixth period to allocate the one truck (step 4).

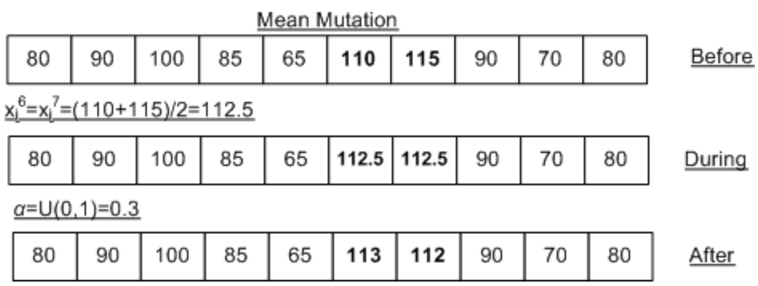


Figure 4. Mean mutation example

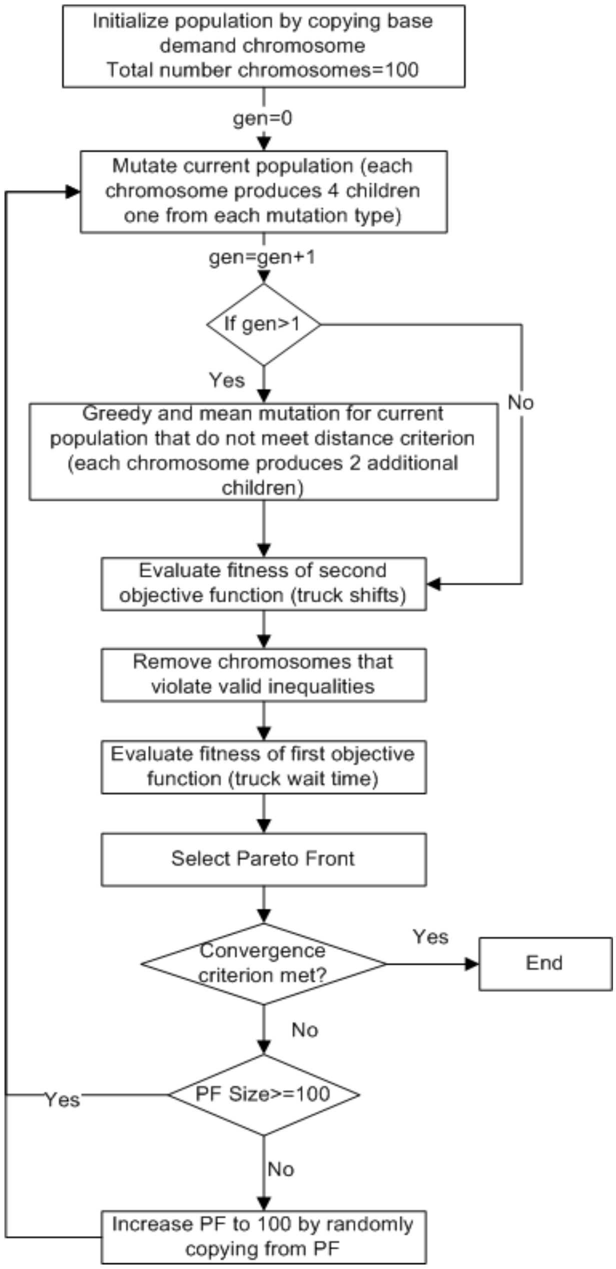


Figure 5. GABH Flowchart

4.2 Valid inequalities

The most time consuming part of the GABH is the evaluation of the first objective function (i.e. the total wait time of all the trucks). To avoid evaluating chromosomes that will not belong to the Pareto Front due to an excess in truck shifts we adopt the following approach. We can easily show[4] that any chromosome with truck shifts greater than those occurring under an evenly spread demand (i.e. the solution with the smallest possible delay) will not belong to the PF. In the proposed GABH, and before we evaluate the first objective function fitness values, we remove chromosomes that do not meet this condition. The full GABH is presented in the flowchart in figure 5.

[4] A solution with an evenly spread demand over all time periods is a lower bound for the total truck waiting time and thus an upper bound for the number of trucks shifts.

4.3 Chromosomal Selection

The selection of the chromosomes that will proceed to the next iteration is based on the Pareto Dominance Criterion. Let *P* denote the list of chromosomes at an iteration of the GABH and let (f_1^l, f_2^l, f_3^l) correspond to the p^{th} chromosome in *P*, where (f_1^l, f_2^l, f_3^l) are the first, second and third objective function values of chromosome *p*. The dominant chromosomes (i.e. the chromosomes that belong to the PF) are obtained as follows:

Step 1: For p=1:|P|
Step 2: For k=p+1:|P|
Step 3: If $f_1^p \leq f_1^k$, $f_2^p \leq f_2^k$, and $f_3^p \leq f_3^k$ set $P := P - k$ and go to step 2
Step 4: Else, if $f_1^p \geq f_1^k$ $f_2^p \geq f_2^k$, and $f_3^p \geq f_3^k$ set $P := P - p$ and go to step 2
Step 5: Return *PF = P*

In Steps 3 and 4 the conditions of the Pareto selection define the dominance relations between the two chromosomes compared. If one of the chromosomes has lower values for both of the objective functions, it dominates the other chromosome, i.e., the other chromosome cannot be in the PF and is omitted from the list. When the PF selection terminates, the reduced list corresponds to the Pareto Front. We note that the worst case complexity of the Pareto Front Selection Routine is $O(n^2)$.

4.5 Stopping criterion

The proposed GABH terminates when the PF does not improve (i.e. solutions in the PF do not change) for 100 iterations.

5 NUMERICAL EXPERIMENT

5.1 Experiment Setting

In this section, we conduct a numerical experiment using the data from Guan and Liu (2009) to test the proposed optimization model. The sample terminal has an entrance gate with six lanes. On a particular day 996 trucks entered the terminal from 06:00 to 16:00, as shown by the gray columns in Figure 10. The truck arrivals showed a pattern of two peaks: one in the morning around 08:00 and the other in the early afternoon. According to Guan and Liu (2009), the truck inter-arrival times follow an exponential distribution, and the gate service times follow an Erlang distribution with a mean of 2.44 min and a parameter (k) of four. We assume five lanes are open in that day, because if six lanes are open, the truck queue will be too short to employ truck arrival coordination. There is no data describing the yard service in Guan and Liu (2009), so we adopt the following data from Kim and Kim (2002): the average service time in each yard is 79 seconds and the variation of the service distribution is 683 seconds. In order to synchronize the yard capacity and the gate capacity, we assume that the number of yard zones is three in the terminal.

5.2 Experiment Result

The GABH is coded using Matlab 7.8 on a PC (Intel T7300 Core 2 Duo). It takes about 6 min to run 1,000 iterations. The obtained result of the proposed multi-objective optimization model is drawn in a 3D graph, as shown in Figure 6 and 7. It clearly indicates that the volume of shifted arrivals has a direct trade-off with both the maximum truck waiting time (Figure 6 -

left) and the total truck waiting time (Figure 6 - right), and there is an obvious liner relationship between the total truck waiting time and the maximum waiting time. Based on this finding, we decide to modify the tri-objective optimization model to a bi-objective one by removing the third objective: the maximum truck waiting time, which can be alternatively estimated with Equation 23. Equation 23 is derived from Figure 7, which has R^2 of 0.9943.

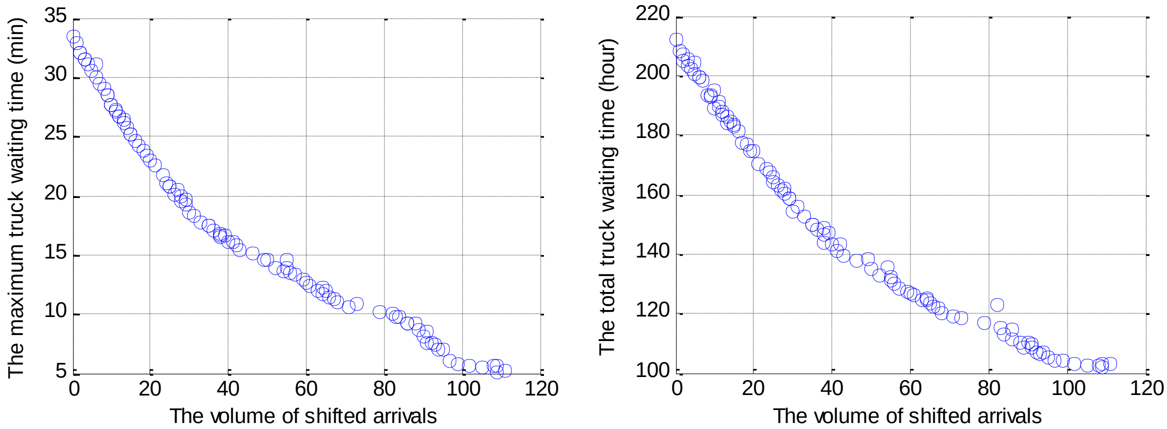


Figure 6. Impact of arrival shifts on the maximum and the total waiting time (tri-objective model)

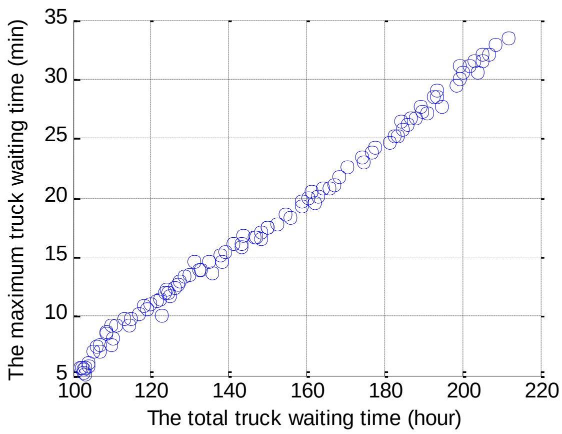


Figure 7. Relationship between the maximum and the total waiting times (tri-objective model)

$$Z^3 = Z^2 \times 0.243 - 18.683 \quad (23)$$

The result of the bi-objective optimization model is shown in Figure 8. The pareto optimal solutions forms a smooth front line, meaning the total truck waiting time could be reduced from 212 hours down to 109 hours at the expense of shifting arrivals. It is interesting to have a close look at the components of the total truck waiting time in Figure 8: as the volume of shifted arrivals in an optimal solution increases, the waiting time at gate is significantly reduced, while the waiting time in yard is only slightly reduced. It indicates that in a container terminal when the gate processing rate and the yard processing rate match to each other, as in this experiment, truck waiting time reduction can be expected mainly at gate rather than in yard, because the departure rate from the gate is relatively steady anyway.

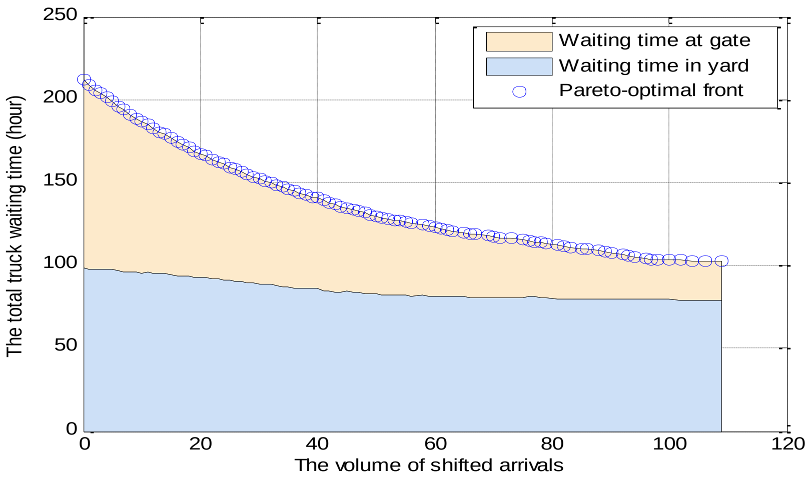


Figure 8. Impact of arrival shifts on truck waiting time at gate and in yard (bi-objective model)

In Phase II, we calculate the emission volumes that can be reduced in these pareto optimal solutions. To better illustrate the impact of shifted arrivals on emission reduction, we show some of the optimal arrival patterns and the corresponding yearly CO_2 emission reduction volumes in Figure 9. We can see: 1) an optimal arrival pattern can be reached by shifting arrivals from peak times to off-peak times; 2) a flatter arrival pattern can achieve more emission reduction. Taking the optimal arrival pattern of 40 shifted arrivals as an example (the blue line marked as 120 in Figure 9), Table 3 shows the reduction volumes of other emissions. This result approves that operation coordination is an effective way to reduce truck idling emission in a container terminal.

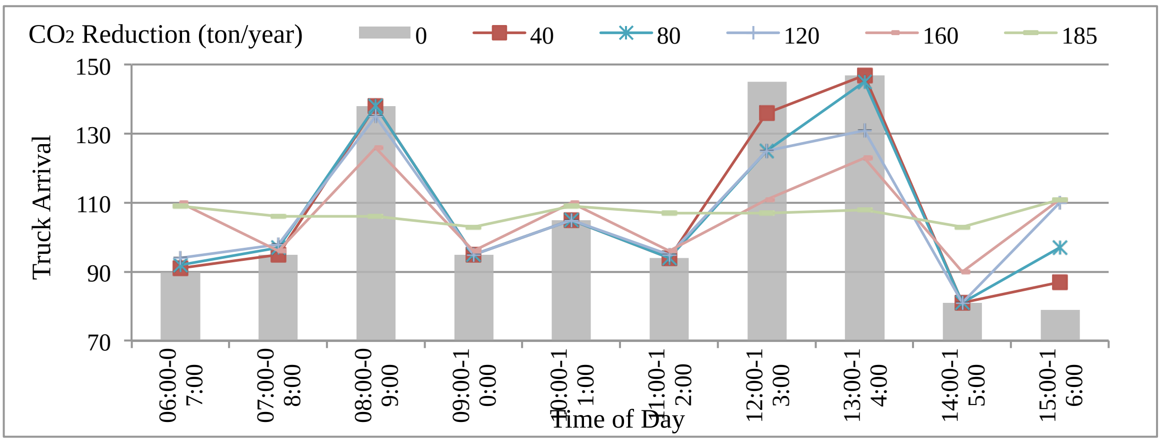


Figure 9. CO_2 reductions can be achieved by optimized truck arrival patterns

Table 3. The yearly emission reduction of the optimal solution of 40 shifted arrivals (kg)

CO_2	N_2O	CH_4	PM_{10}	$PM_{2.5}$	DPM	NO_X	SO_X	CO	HC
119,687	1	5	6	5	5	2,444	1	434	161

6 CONCLUSIONS

This study proposes a queueing network based multi-objective model to investigate the trade-off relationship between truck idling emissions and truck arrival pattern at a container terminal. We develop a two-layer queueing network to model the truck queueing processes both at gate and in yard, A multi-objective optimization model is developed to investigate the trade-off between truck idling emissions and truck arrival pattern. A GA based heuristic is developed to solve the model. Numerical experiment shows that, truck emissions could be significantly reduced by shifting a small number of truck arrivals. This finding indicates that it is essential to employ truck arrival coordination in a container terminal to improve the system efficiency and to reduce port-related emissions.

REFERENCES

Chen, G., & Govindan, K. (2011). A Multi-Objective Based Green Logistics Model to Reduce Truck Emissions at Container Terminals. *the First International workshop on Eco-Efficient Based Green Supply Chain Management – EF-GSCM 2011*. Odense, Denmark.

Chen, G., & Yang, Z. Z. (2010). Optimizing time windows for managing arrivals of export containers at Chinese container terminals. *Maritime Economics & Logistics*, Volume 12, Pages 111–126.

Chen, G., Govindan, K., & Yang, Z. Z. (2011a). Designing terminal appointment system with integer programming and non-stationary queueing model. *Technique Report, Univsersity of Southern Denmark*.

Chen, X., Zhou, X., & List, G. F. (2011b). Using time-varying tolls to optimize truck arrivals at ports. *Transportation Research Part E: Logistics and Transportation Review*, Volume 47, Pages 965-982.

Cosmetatos, G. P. (1976). Some Approximate Equilibrium Results for the Multi-Server Queue (M/G/r). *Operational Research Quarterly*, Volume 27, Issue 3, Pages 615-620.

Goodchild, A., & Mohan, K. (2008). The Clean Trucks Program: Evaluation of Policy Impacts on Marine Terminal Operations. *Maritime Economics & Logistics*, Volume, Pages 393-408 .

Green, L., & Kolesar, P. (1991). The pointwise stationary approximation for queues with nonstationary arrivals. *Managemetn Science*, Volume 37, Issue 1, Pages 84-97.

Guan, C. Q., & Liu, R. f. (2009). Container terminal gate appointment system optimization. *Maritime Economics & Logistics*, Volume 11, Issue 4, Pages 378–398.

Huynh, N. N. (2009). Reducing Truck Turn Time at Maritime Container Terminals with Appointment Scheduling. *In Transportation Research Record: Journal of the Transportation Research Board*, Issue 3230, Pages 48–56.

Huynh, N., & Walton, C. M. (2008). Robust Scheduling of Truck Arrivals at Maritime Container Terminals. *Journal of Transportation Engineering*, Volume 134, Issue 8, Pages 347-353.

Huynh, N., Harder, F., Smith, D., Sharif, O., & Pham, Q. (2011). Truck Delays at Seaports - Assessment Using Terminal Webcams. *Transportation Research Record: Journal of the Transportation Research Board*, Issue 2222, Pages 54–62.

Huynh, N., Walton, M. C., & Davis, J. (2004). Finding the number of yard cranes needed to achieve desired truck turn time at maritime container terminals. *In Transportation Research Record: Journal of the Transportation Research Board*, Issue 1873, Pages 99–108.

Kim, K. H., & Kim, H. B. (2002). The optimal sizing of the storage space and handling facilities for import containers. *Transportation Research Part B: Methodological*, Volume 36, Pages 821-835.

Meyer, P. D., Maes, F., & Volckaert, A. (2008). Emissions from international shipping in the Belgian part of the North Sea. *Atmospheric Environment*, Volume 42, Pages 196-206.

Saxe, H., & Larsen, T. (2004). Air pollution from ships in three Danish ports. *Atmospheric Environment*, Volume 38, Pages 4057-4067.

Smith, J. M. (2010). Robustness of state-dependent queues and material handling systems. *International Journal of Production Research*, Volume 48, Page 4631–4663.

Starcrest Consulting Group, L. (2011). *The Port Of Los Angeles Inventory of Air Emissions For Calendar Year 2010.* Los Angeles: Port of Los Angeles.

Tzannatos, E. (2010). Ship emissions and their externalities for the port of Piraeus. *Atmospheric Environment*, Volume 44, Page 400-407.

Wang, W.-P., Tipper, D., & Banerjee, S. (1996). A simple approximation for modeling nonstationary queues. *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, 255 - 262 vol.1 .

Whitt, W. (1991). The pointwise stationary approximation for Mt/Mt/s queues is asymptotically correct as the rates increase. *Management Science*, Volume 37, Pages 307-314.

Zietsman, J., Farzaneh, M., Schneider-IV, W. H., Lee, J. S., & Bubbosh, P. (2009). Truck Stop Electrification as a Strategy To Reduce Greenhouse Gases, Fuel Consumption and Pollutant Emissions. *Transportation Research Board Annual Meeting 2009.* Washington D.C.: Transportation Research Board.

DYNAMIC DETERMINATION OF VESSEL SPEED AND SELECTION OF BUNKERING PORTS FOR LINER SHIPPING UNDER STOCHASTIC ENVIRONMENT

Xiaoming Sheng[1], Loo Hay Lee[1] and Ek Peng Chew[1]

[1]*Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, 119260, Singapore, Sheng@nus.edu.sg*

ABSTRACT

In this work, we study a liner shipping operational problem which considers how to dynamically determine the vessel speed and refuelling decisions, for a single vessel in one service route. Our model is a multi-stage dynamic model, where the stochastic nature of the bunker prices is represented by a scenario tree structure. Also, we explicitly incorporate the uncertainty of bunker consumption rates into our model. As the model is a very large scale mixed integer programming model, we adopt a modified rolling horizon method to tackle the problem. Numerical results show that our framework provides a lower overall cost and more reliable schedule compared with the stationary model of a related work.

Keywords: Bunker Price, Vessel speed, Scenario tree generation, Rolling horizon

1 INTRODUCTION

In recent years the trend of increasing bunker prices has threatened the liner shipping companies' accounting bottom line. To survive, companies need to identify ways to reduce the operating costs. For example, when the oil prices hit $145 a barrel in 2008, Maersk, the world's biggest liner shipping company, spearheaded the strategy of slow steaming. Now over two hundred shipping companies has reduced their vessel speeds, especially in those long-haul loops like Asia to Europe and North America. Empirical estimation has shown that when the vessel speed is reduced by 20%, it could reduce the fuel consumption by 50%. Besides, the environmental benefits of less greenhouse gas emission from slow steaming are also significant. Maserk (2009) announced that on average they had successfully reduced the carbon dioxide emission by 14% per vessel during 2008.

Besides increasing rapidly, bunker prices also manifest high volatility. It is a well observed phenomenon that the crude oil prices fluctuate significantly on a daily basis. As a by-product of the crude oil, bunker's prices fluctuate no lesser in the spot market. Figure 1 below shows the monthly fluctuation of bunker prices (380 CST grade) at several major ports and that of crude oil prices from Sep-02 to Sep-09. Based on this figure, we can roughly say that there is a high correlation of the bunker prices and crude oil prices and most of the time, the bunker prices are even more volatile than the crude oil prices. Last but not least, bunker prices at different ports around the world usually have significant differences. For example, on September, 3, 2008, bunker fuel prices (380 CST) in Singapore were 677.5 US$/ton. On the same day, bunker prices in Rotterdam were 619 US$/ton and 650 US$/ton in Houston.

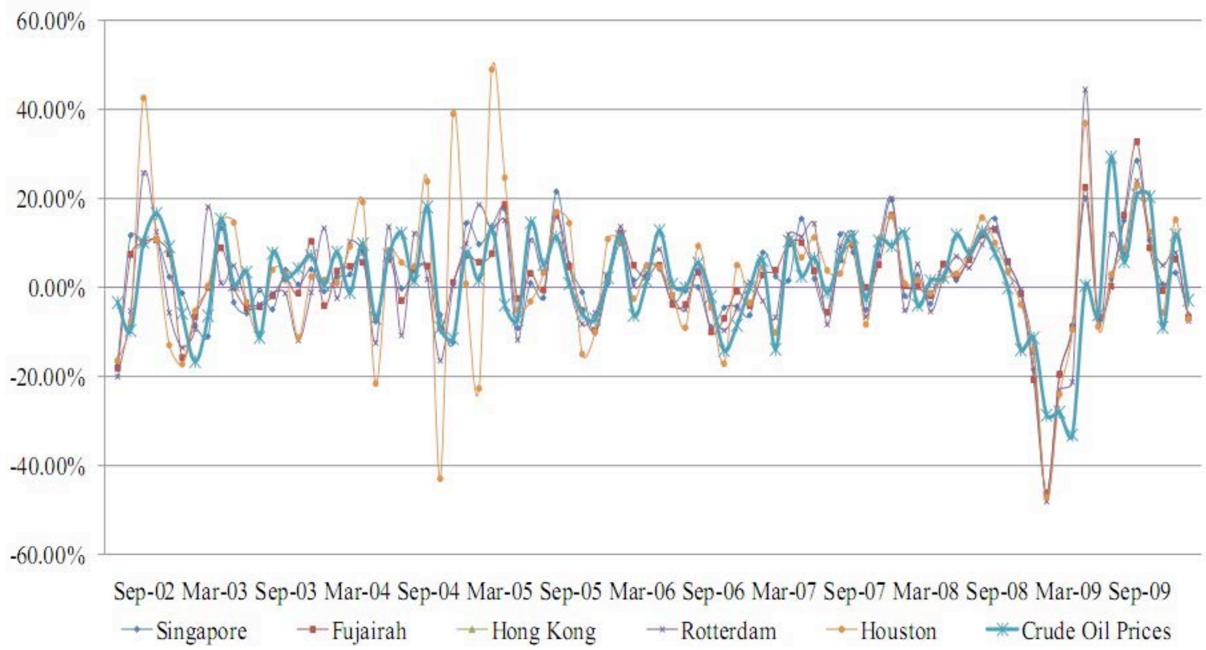


Figure 1. Fluctuation of the bunker fuel (380 CST) prices at major bunkering ports and the world Crude oil prices (2002-2009) (Data source: Bloomberg 2009 and www.test.org/doe/)

The characteristic of the liner shipping is that it usually has a fixed number of port-calls in a cyclical route with a published schedule. While slow steaming would be the general trend when bunker prices are high, high fluctuation and regional differences of the bunker prices complicate the situation because simply reducing the vessel speed may miss out the opportunity of reaching the next port when bunker prices there are low. Thus, in this work, we study how to dynamically determine the vessel speed from the current port to the next one and how much fuel to bunker in the current port with all the bunker prices and bunker consumption information available so far for a single vessel in one service route. The case of multiple vessels can be easily extended. This is an operational level problem when, on the planning level, ships deployment, scheduling and routing have been decided.

Most of the previous related works did not tackle the uncertain nature of this problem sufficiently. Ronen (1982) studied the trade off between fuel savings from slow steaming and loss of revenue from extension of voyage. It approximated the daily bunker consumption as a third power of ship speed and derived the optimal speed for ships under different operating scenarios, namely, income generating leg, positioning leg and mixed leg. Notteboom et al. (2009) studied how liner shipping, facing high bunker fuel prices, had adapted their liner service schedules. In this study, the authors provided real world data about fuel consumption per day and ship speed for different size of container ships. From the data shown, we can see that the fuel consumption rate against speed for different sizes of ships is actually different although the authors did not look into the details of this issue. Ronen (2011) investigated into the trade off between slow steaming and adding additional vessels in a container route. The objective was to minimize the annual operating cost of the route by deciding the optimal vessel speed and number of ships deployed.

Yao et al. (2011) studied a problem similar in nature with ours, but in their study, the focus was put on the planning level problem, thus no uncertainty was addressed. On the relationship between the bunker consumption rate and ship speed, they separated the analysis of it by the different sizes of ships. In addition, instead of assuming a single third power relationship, they added a constant coefficient in the regression model, which they proved to be non-trivial by numerical experiments. In all of those above studies, bunker prices were either assumed to be constant or not explicitly considered.

To the best knowledge of authors, there is no published result that considers the bunker consumption uncertainty. However, wind force and direction, sea condition, engine efficiency and other factors can influence the bunker consumption in a significant basis. Another main reason is that we use the average ship speed between each leg in our problem, while, in reality, the ship speed could vary within each leg. Only by considering the randomness of consumption rate at any speed could we capture the real world scenarios more precisely and provide more reliable operational level recommendations for liner shipping practitioners. Therefore, our work is the first attempt to tackle the speed determination and refuelling decision simultaneously with the consideration of bunker prices and bunker fuel consumption rate uncertainties.

We will formulate our problem as a multi-stage dynamic model, where the randomness of the bunker prices is represented by a scenario tree structure. As the model is a very large scale mixed integer programming model, we adopt a modified rolling horizon method to tackle the problem.

The rest of the paper is organized as follows: In the following section, we will give a general description of our problem. Also modelling for the uncertain bunker prices and daily consumption rate will be discussed. In Section 3 and 4, we present our dynamic model and rolling horizon solving approach. One case study will be conducted in Section 5. Finally, the conclusion would be presented in Section 6.

2 DESCRIPTION OF THE PROBLEM

In this paper, we consider the operational level decision making for a single liner ship in one cyclical route (start from one port, travel through all other ports at least once and go back to the original port) with fixed number of port-calls and time windows. Time window states the ship arrival and departure times at each port. Two uncertain factors considered in our work are the bunker fuel prices and bunker fuel consumption rate. A more detailed discussion about how to capture the randomness of bunker prices and the stochastic nature of bunker fuel consumption rate will be given later in this section.

Two key decisions to be made are where to bunker and what is the bunker-up-to level. We assume that bunkering can only happen when ship reaches one port. Bunkering decision depends on the bunker prices at each port which are usually different across those ports due to local supply-demand factors. The evolution of bunker prices at each port can be modeled as a discrete-time Markovian process which describes all the possible states and transition matrix between those states. Without loss of generality, we assume that port calls are on a weekly basis and hence we only need to describe the bunker prices evolution on a weekly basis. While this is a drawback of our work, our modified rolling horizon approach can help to mitigate this problem. Because, we can always update the bunker price scenario tree based on timely real world situation.

Aside from bunkering, another important decision is the ship speed between each leg, which has been commonly assumed to be constant during each leg. How to reach each port within the scheduled arriving time and save bunker consumption as much as possible through slow steaming is a question faced by most of the practitioners.

In our problem, the objective is to minimize the total operational cost in one service loop. The costs considered here are the bunker cost and inventory holding cost. Bunker cost mainly consists of two parts, fixed bunkering cost incurred each time a bunkering takes place and variable cost that depends on the bunkering amount and bunker prices at the time being. Inventory holding cost can be interpreted as a combination of capital committed in the bunker purchase which could otherwise generate profits through some investment activities and a loss of revenue due to less capacity to carry revenue-generating cargo. As a simplification, we assume that the inventory carrying cost pmt (per metric ton) is constant. Because our study horizon is one service loop which is finite, inevitably, there would be bunker fuel left in the ship fuel tank at the end of voyage. For this amount of bunker fuel left, we deduct it from the total cost based on the bunker prices at the time being.

2.1 Model for bunker prices

To model the evolution of the bunker prices, we use the percentage changes in each leg of the voyage, but the difficulty is that the percentage change can take any continuous value within a reasonable range. Incorporating a random variable with continuous distribution into an optimization model would make solving the model extremely hard, if not impossible.

Therefore, we discretize the bunker price percentage changes. At first, we determine an interval within which the bunker price percentage changes between two subsequent periods can take and then we divide this interval into several smaller sub-intervals. Transition matrix depicts the transition among those sub-intervals is constructed. In the end, one discrete point value is chosen to represent each sub-interval. We can either choose the mean of the sub-interval or generate it by random sampling.

2.2 Model for bunker consumption rate

In Yao et al. (2011), they assumed that the daily bunker consumption rate could be expressed as: $F = k_1.V^3 + k_2$, within which k_1 and k_2 are two constants and they can be different for different vessel sizes, and V is ship speed (knots). Due to reasons we mentioned beforehand, a noise term is added in order to depict the uncertainty of bunker consumption. This means:

$$F = k_1.V^3 + k_2 + \varepsilon(V)$$

Based on the data we obtained, we found that the noise ε is a function of the ship speed and the noise term follows a normal distribution with zero mean and constant coefficient of variation under different ship speeds. This means that the standard deviation of bunker consumption within a specific period of time is proportional to the mean consumption. Remembering that wind and sea current are two of the main factors for the bunker consumption uncertainty and the fact that their influence increases with ship speed would not surprise us with such a conclusion. Also, for different sizes of ships, we found that the CV actually decreases with ship size. The intuitive explanation is that bigger ships are usually equipped with more powerful engines, and thus wind and sea current impose relatively less influence on them.

In our dynamic model, we will use chance constraints to control the probability of running out of bunker before reaching the next port to be less than one critical value.

3 MODELLING

As discussed, we use the percentage changes in each leg of the voyage to model the evolution of the bunker prices.

In the financial engineering area, researchers use scenario tree models to capture the randomness of financial products, interest return, bonds, stocks, etc. Mulvey et al. (1997) reviewed the application of multi-stage stochastic optimization on asset/liability management. When it came to the trade-off between realism and computational tractability, they listed several essential characteristics that a mathematical model for investment problem should possess in order to render useful application. One possible way that they claimed to be effective in covering all of those characteristics was scenario tree model. Considering the similar nature of those financial products with bunker prices, here we use the scenario tree to model the randomness of bunker prices.

Bunker prices uncertainty in the future times is modeled by a discrete stochastic process ξ that is defined on a probability space of (Ω, F, P) with

$$\xi = \{\xi := \theta_t^i\}_{t \in T}.$$

Here P_t^i and θ_t^i denotes the bunker prices and bunker price percentage changes at port *i* and time period *t*. There are baseline bunker prices P_0^i at port *i* and time 0. Therefore, bunker prices at each port and time period *t* are based on baseline bunker prices as well as all the percentage changes during previous periods. For example, $P_1^i = P_0^i \times \theta_1^i$ and $P_k^i = P_{k-1}^i \times \theta_k^i$.

At the beginning of every service loop, the most recent bunker price changes are known. For the future bunker price changes, we only know the discrete probability distribution. And bunkering and speed decisions at any stage do not depend on future realizations of bunker price changes, but on the probability specification (Ω, F, P). This is a non-anticipative constraint commonly used in many multi-stage stochastic optimization problems. When it comes to the end of studying horizon, all the random information is realized. A series of realizations $(\xi_1^s, ..., \xi_T^s)$ over the entire study horizon consist of a scenario. All the scenarios are combined into a scenario tree representation. Figure 2 below shows an example of scenario tree. As mentioned, in our work, we approximate the port-by-port bunker price change evolution by weekly bunker price change evolution.

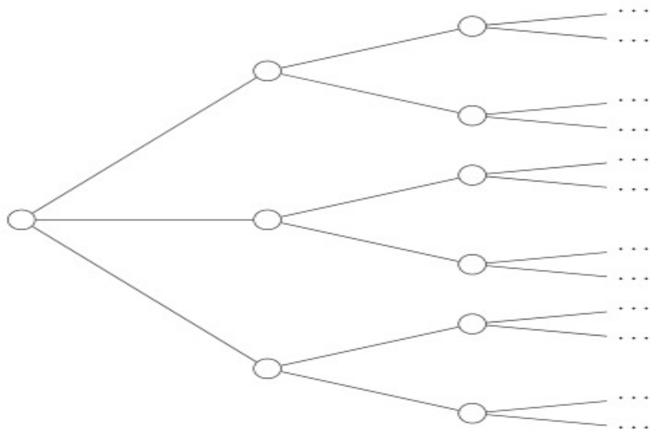


Figure 2. A simple example of scenario tree

3.1 Assumptions

Now, we summarize all the assumptions made in our paper:

1. We consider one ship in one service route with time windows. Port time (time one ship spends on entering, unloading and loading cargo, idling and exiting) at each port is deterministic and known.
2. Bunkering and ship speed decisions are made and only made when ship reaches one port.
3. Relationship between the ship speed and bunker consumption is established in Section 2.
4. Bunker prices at different ports are not necessary the same. In addition, bunker price change follows a discrete-time Markovian process.
5. Bunkering cost includes the fixed cost which is constant over time by assumption and variable cost. Bunker inventory cost pmt (per metric ton) is assumed to be constant and independent of bunker prices. Bunker left at the end of one service loop is refunded.

3.2 Notations

Parameters

n	number of port of calls;
S	total number of price scenarios;
Π^s	the probability that price scenario s happens;
$d_{i,j}$	distance between port i and port j (nautical miles);
t	total cycle time (hours);
t_i	port time (time one ship spends on entering, unloading and loading cargo, idling and exiting) at port i (hours);
e_i	earliest arrival time at port i (hour);
l_i	latest arrival time at port i (hour);
c_i	bunker fuel consumption when the ship is at port i;
w	bunker fuel capacity of a single ship (tons);
v_{min}	minimum ship speed (knots: nautical miles/hour);
v_{max}	maximum ship speed (knots);
k_1, k_2	bunker fuel consumption coefficients;
P_i^s	bunker price for port i under scenario s (\$/ton);
f	fixed bunkering cost;
γ	coefficient to control the service level;
h	inventory holding cost for bunker;
η	coefficient of variation for daily bunker consumption rate;

Decision variables

$V_{i,j}^s$	ship speed between port i and j under scenario s (knots);
R_i^s	bunker fuel-up-to level for the ship at port i under scenario s (tons);
B_i^s	bunkering decision variable. =1 if bunkering at port i under scenario s; =0, otherwise;

Dependent variables

I_i^s	bunker fuel inventory when the ship reaches port i under price scenario s;
$\bar{F}_{i,j}^s$	mean of daily bunker consumption rate for a ship travels from port i to j under price scenario s (tons/day);
$\delta_{i,i+1}^s$	standard deviation of bunker fuel consumption between port i and $i+1$ under price scenario s (tons);
D_i^s	standard deviation of ship bunker inventory when ship reaches port I under price scenario s(tons);
A_i^s	ship arrival time at port i under scenario s (hour);

3.3 Mathematical model

We present a mathematical model to describe our problem.

$$\text{Min} \quad \sum_{s=1}^{S} \Pi^s \left(\sum_{i=1}^{n} [(R_i^s - I_i^s) P_i^s + f.B_i^s + (R_i^s - c_i^s).h] - I_{n+1}^s.P_{n+1}^s \right)$$

Subject to

$$I_1^s = 0, \qquad D_1^s = 0 \qquad\qquad \forall s \in S \qquad (1)$$

$$I_i^s = R_{i-1}^s - c_{i-1}^s - \bar{F}_{i-1,i}^s \cdot d_{i-1,i} \big/ (24 \cdot V_{i,i-1}^s) \qquad\qquad \forall s \in S\ , i \in 2,3,\ldots,n+1 \qquad (2)$$

$$R_i^s - I_i^s \le B_i^s.w \qquad\qquad \forall s \in S\ , i \in 1,2,\ldots,n \qquad (3)$$

$$R_i^s \leq w \qquad \forall s \in S\ , i \in 1,2,...,n \qquad (4)$$

$$\delta_{i-1,i}^s + (1 - B_{i-1}^s).D_{i-1}^s = D_i^s \qquad \forall s \in S\ , i \in 2,3,...,n+1 \qquad (5)$$

$$\bar{F}_{i,i+1}^s = k_1 \cdot (V_{i,i+1}^s)^3 + k_2 \qquad \forall s \in S\ , i \in 1,2,...,n \qquad (6)$$

$$\delta_{i-1,i}^s = \eta \times \bar{F}_{i-1,i}^s .d_{i-1,i} / (24.V_{i-1,i}^s) \qquad \forall s \in S\ , i \in 2,3,...,n+1 \qquad (7)$$

$$I_i^s \geq \gamma \times D_i^s \qquad \forall s \in S\ , i \in 2,3,...,n+1 \qquad (8)$$

$$v_{\min} \leq V_{i,i+1}^s \leq v_{max} \qquad \forall s \in S\ , i \in 1,2,...,n \qquad (9)$$

$$A_i^s + t_i + d_{i,i+1} \big/ V_{i,i+1}^s = A_{i+1}^s \qquad \forall s \in S\ , i \in 1,2,...,n \qquad (10)$$

$$e_i \leq A_i^s \leq l_i \qquad \forall s \in S\ , i \in 1,2,...,n \qquad (11)$$

$$A_{n+1}^s = t \qquad \forall s \in S \qquad (12)$$

$$B_i^s = 0 \text{ or } 1 \qquad \forall s \in S\ , i \in 1,2,...,n \qquad (13)$$

$$V_{i,i+1}^s = V_{i,i+1}^{\bar{s}} \qquad \forall (s,\bar{s}) \in S \text{ identical past to } i+1, i \in 1,2,...,n \qquad (14)$$

$$R_i^s = R_i^{\bar{s}} \qquad \forall (s,\bar{s}) \in S \text{ identical past to } i, i \in 1,2,...,n \qquad (15)$$

$$B_i^s = B_i^{\bar{s}} \qquad \forall (s,\bar{s}) \in S \text{ identical past to } i, i \in 1,2,...,n \qquad (16)$$

$$F_{n,n+1} = F_{n,1},\ d_{n,n+1} = d_{n,1},\ V_{n,n+1}^s = V_{n,1}^s . \qquad \forall s \in S \qquad (17)$$

The objective function is to minimize the expected total cost, which includes the fixed and variable bunkering cost and inventory holding cost. Bunker inventory left at the end of one service loop or beginning of a new loop is deducted as monetary terms. Constraint (1) sets the initial ship bunker inventory and standard deviation of it at zero. Constraint (2) is flow conservation constraint. Constraint (3) and (4) ensure that the maximum bunkering amount and bunker-up-to level is less than the fuel tank capacity. Constraint (5) states that if the ship bunkered at previous port, then standard deviation of the ship bunker inventory at current port is equal to the standard deviation of bunker consumption from previous port to the current port. Otherwise the standard deviation of ship bunker inventory at previous port should also be added. This is because, as discussed, standard deviation of bunker consumption is proportional to the total bunker consumption. Constraint (6) and (7) express the mean daily consumption rate at a certain speed and stand deviation of bunker consumption between ports i and i + 1. Constraint (8) is the deterministic equivalent for chance constraint $P\{I_i^s \geq D_i^s\} \geq \gamma$ which ensures that the probability of bunker inventory being greater than a certain amount is greater than a pre-specified value. Constraint (9) is simply to limit the ship speed within a reasonable range, while constraints (10) to (12) are about time window constraints. Constraint (13) is a binary constraint. Constraints (14) to (16) are non-anticipative constraints which ensure that scenarios that share the same history up to port i should take the same action.

4 SOLUTION METHOD

There are two potential challenges in solving our problem. The first one is the non-linearity constraints related to the ship speed. We deal with this by following the method used in Yao et al (2011), which applied a piece wise linear function to approximate the non-linear terms.

Another challenge is that when a scenario tree procedure is used to model the stochastic parameters in a multi-stage stochastic problem, solving the problem is usually difficult because of the large problem size. For example, in a case where there are 15 ports and for each period (ship reaches a new port) there are four price scenarios, the total number of scenarios in a scenario tree construction would be 4^{16}.

In this work, we propose to use a slightly different method of generating scenario tree, and combine it with a modified rolling horizon approach to solve a liner shipping operational level problem. The rationale behind this combination is: firstly, bunker prices forecasting which covers a long period of time, if not impossible, suffers greatly in terms of forecasting quality. Instead of making one time forecast only at the very beginning for the whole horizon, constantly updating the forecast and resolving the optimization problem are beneficial; secondly, this successfully circumvents the trouble of solving a large scale stochastic optimization problem.

4.1 Rolling horizon planning scheme

The essence of the standard rolling horizon planning scheme is: A problem with the study horizon shorter than the original one (to reduce the problem size) is solved and the first period decision is implemented. With newly available information, the problem is updated and resolved. Still the decision is only acted on for the imminent period. This process goes on and on until the end of the study horizon. For example, Baker (1977) implemented the standard rolling horizon approach in a production planning problem and numerical results in his work showed that rolling horizon approach produced results that were well within 10% of optimality and if the model construction was well tailored for specific circumstance, the optimality gap could be further reduced within 1%. In addition, he mentioned two key reasons ("uncertain information about the future" and "limited information about the future") that legitimized the use of finite-horizon model for the purpose of decision making in infinite-horizon system.}

In our case, we will use a non-standard rolling horizon approach. Unlike the standard one which solves a problem with a shorter horizon than the original problem, our non-standard approach still solves the problem with the whole study horizon. However, we assign a higher level of fidelity for the nearer periods than the later ones by modifying the way we generating the scenario tree. For the first few number of periods (could be 1,2 or any number of periods depending on the problem), all the price change alternatives are generated as shown in Figure 2, while a relatively small number of realizations (also problem specific) are randomly generated for all the remaining periods till the end. Therefore, an example of our modified version of scenario tree would look like Figure 3. The validity of this non-standard variant is due to our problem nature and the diminishing tail-end effect. We will further show the suitability of using this non-standard solving horizon approach through our first numerical example. }

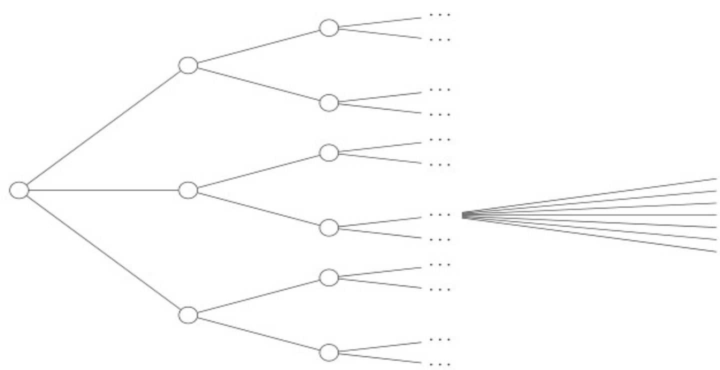


Figure 3. Scenario tree with randomly generated siblings

5 A CASE STUDY

Here, we implement our model in one real-world service routes, Asia-Europe Express (AEX), offered by a liner shipping company. AEX route has 15 port-of-calls. We use the rolling horizon approach to solve it and compare the results with that of the stationary model in Yao et al. (2011). However, we have modified their model in order to make a fair comparison.

5.1 Parameter setting for bunker price changes

We construct three cases of weekly bunker price percentage changes as denoted by Case 1, 2 and 3. For example, Table 1 and Table 2 list the weekly bunker price change alternatives and transition matrix for Case 1. Weekly bunker price change alternatives for Case 2 and Case 3 are [-15%,-7.5%, 7.5%, 15%] and [-20%,-10%, 10%, 20%] respectively. And all three cases share the same transition matrix. The rationale behind the constructing of those three cases is that our work is more relevant in times when bunker prices are highly fluctuating (September 2008, for example, IFO380 averaged slightly over $600 pmt in Singapore, however it dropped to average $410 in October).

Table 1. Weekly bunker price change alternatives: Case 1

Scenario	Value	Probability	Value
C_1	-10%	$P(C_1)$	0.25
C_2	-5%	$P(C_2)$	0.25
C_3	5%	$P(C_3)$	0.25
C_4	10%	$P(C_4)$	0.25

Table 2. Transition Matrix of the weekly bunker price changes: Case 1

Scenario	C_1	C_2	C_3	C_4
C_1	40%	30%	20%	10%
C_2	30%	40%	20%	10%
C_3	10%	20%	40%	30%
C_4	10%	20%	30%	40%

5.2 AEX service route

AEX service route consists of 15 ports which means there are altogether 4^{16} scenarios and parameter setting is given below and also the same with that in Yao et al. (2011) for the purpose of fair comparison. In this example, we are going to solve the problem using the modified rolling horizon approach and then compare the results with the stationary model.

Parameter setting for AEX service route is given below:

Table 5. Parameter setting for AEX service route

Parameter	Value
Number of port of calls	15
Service frequency	weekly
Ship size	6000TEU
Total cycle time	1512 hours
Ship speed interval	14--24 knots
Mean bunker consumption rate	$F = 0.007297 V^3 + 71.4$
Coefficient of variation of bunker consumption rate	0.07

Failure rate in both models is controlled to be 0.01. And in the modified rolling horizon method of this example, we look ahead 3 ports which we fully generate all the alternatives for them and for the remaining 12 ports, 8 price realizations are generated. In our comparison, 40 price scenarios have been generated. Under Case 1 of the bunker price changes setting, average cost for the dynamic model solved by the modified rolling horizon approach is $3,660,000 and average cost for the stationary model is $3,840,000 which is about 4.9% of cost saving. Under Case 2 of the bunker price changes setting, average cost for dynamic model solved by the modified rolling horizon approach is $3,790,000 and average cost for the stationary model is $4,070,000 which is about 7.4% of cost saving. Under Case 3 of the bunker price changes setting, average cost for dynamic model solved by the modified rolling horizon approach is $3,820,000 and average cost for the stationary model is $4,300,000 which is about 12.6% of cost saving.

From Case 1 to Case 3, as the bunker prices become more volatile, the cost saving of the modified rolling horizon approach compared with the stationary model increases from 4.9% to 12.6%. Considering the huge amount of operational costs for a liner shipping company, this means a significant total cost reduction. Also this result does not surprise us because our dynamic model should become more superior to the stationary one when prices are more fluctuating.

Bunker inventory holding cost pmt in our problem is assumed to be constant and independent of the bunker prices. Thus we want to see how sensitive the result is to this parameter. In this AEX route example, bunker prices of all the ports at the initial stage are set around $460 pmt with minimum $456 and maximum $471. Our previous results are based on bunker inventory holding cost being $50 pmt. In our subsequent analysis, we want to see what will happen if we vary this parameter.

Take Case 1 of the bunker price changes setting for example, our numerical results show that when bunker inventory cost is $100 pmt, rolling horizon approach has 8.56% of cost saving to deterministic approach, compared with 4.9% if bunker inventory cost is $50 pmt. And when inventory cost is $150 pmt, this cost saving increases to 13.4%. Or if we set inventory cost to be $25 pmt, the cost saving is 3.48%. This means generally when bunker inventory cost pmt increases, the dynamic model becomes even more superior to the stationary model.

6 CONCLUSIONS

In this work, we study the problem of dynamic bunkering ports selection and ship speed determination for a single vessel in one service route. While previous deterministic works focus more on the planning level of this problem, we aim at providing operational level decision support by incorporating two major random factors into our model. Namely, the ship bunker consumption rate and the bunker prices at each port. Based on the bunker consumption model in Yao et al. (2011), we further established that the noise of daily bunker consumption follows a normal distribution with zero mean and constant coefficient of variation. For the bunker prices, we have modeled it through the scenario tree which is widely used in financial engineering area to depict the randomness of financial product returns. While solving a whole large dynamic problem generated by a scenario tree is computational challenging, we proposed a solving method that could help to significantly reduce the computer memory requirement and solving time. This method is a combination of scenario tree generation scheme and a modified rolling horizon approach. Another advantage of our work is that as much newly information as possible is used and previous forecasting errors can be easily correctly in the whole study horizon. Our numerical examples based on real world data has shown that the dynamic model improves significantly in terms of overall cost and service level (or failure rate) compared with the stationary one. With the reasonable solving time, we think our model could be implemented by liner shipping companies to give operational decision support in order to lower the overall operation cost and provide more reliable service.

REFERENCES

Baker, K. (1977). An experimental study of the effectiveness of rolling schedules in production planning. *Decision Sciences* , *8* (1), 19-27.

Maersk. (2011). http://www.maersk.com. Accessed on Feb.15.2011.

Mulvey, J., Rosenbaum, D., & Shetty, B. (1997). Strategic financial risk management and operations research. *European Journal of Operational Research , 97* (1), 1-16.

Notteboom, T., & Vernimmen, B. (2009). The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography , 17* (5), 325-337.

Ronen, D. (2010). {The effect of oil price on containership speed and fleet size}. *Journal of the Operational Research Society , 62* (1), 211-216.

Ronen, D. (2011). The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society , 62* (1), 211-216.

Ronen, D. (1982). The effect of oil price on the optimal speed of ships. *The Journal of the Operational Research Society , 33* (11), 1035-1040.

Yao., Z., Ng., S. H., & Lee., L. h. (2011). A study on shipping liner service plans with ship speed and bunkering port considerations. *Computers & Operations Research, 39(5)*, 1160-1172.

ENVIRONMENTAL IMPACT OF SHIPS IN PORTS AND POSSIBLE MITIGATION TECHNIQUES

Thalis Zis[1], Robin J. North[2], Michael G. H. Bell[2] and Panagiotis Angeloudis[2]

[1]*Port Operations Research and Technology Centre, Department of Civil & Environmental Engineering, Imperial College London, London, SW7 2BU, United Kingdom, thalis.zis09@imperial.ac.uk*

[2]*Centre for Transport Studies, Department of Civil & Environmental Engineering, Imperial College London, London, SW7 2BU, United Kingdom, r.north@imperial.ac.uk, mghbell@imperial.ac.uk, p.angeloudis@imperial.ac.uk*

ABSTRACT

In recent years a major concern in all transport operations has been the quantification and management of environmental impacts. However, the air emissions that originate from the shipping sector are still increasingin absolute terms and are expected to continue to do so. In supply chains, vessels often have to be stationed at ports for extended periods of time, and therefore an important part of the shipping sector's emissions is generated in ports. The port is considered as a system comprising several different parts, all of which contribute to its environmental footprint. This paper presents a framework for examining these impacts with a particular focus on the environmental footprint of ships approaching and berthing at ports, where their emissions impact directly on local air qualityand on the exposure of the local population. Speed reduction of approaching ships and cold ironing for berthing ships are examined as potential mitigation techniques, and their scope for the reduction of emissions for CO_2,SO_2 and $NO_x$are assessed.A case study using data from the port of Felixstowe and the UK National Grid is then used to compare the impact of these two techniques on local air quality and onemissions of global pollutants.

Keywords: ship emissions, slow steaming, cold ironing, port operations

1 INTRODUCTION

Ports, as all major transport hubs, have a significant environmental footprint. Ships account for the 3.3% of global CO_2 emissions (IMO 2009). Part of this is generated while vessels arrive and depart to a port andduring berth. In-port operations require energy to support a range of activities, from buildings' lighting to cargo movement. The energy demand varies with size, location, equipment used and predominantly port type (container terminals, bulk terminals, Ro-Ro ports). Road and rail transportation of goods (container, bulk) or people to and from the port also generate significant emissions that are a result of the port's operation.

The impacts of the port operations on the surrounding area can be attributed to three main parts; Maritime operations, in-port operations and generated traffic outside the port's gates. This study focuses on the first part; The maritime operations near and in the port.In order to estimate pollutant emissions from vessels it is necessary to differentiate between ships that travel near the port area and stationary ships as in most cases the latter are using their auxiliary engines with different fuel consumption, or connected to shore power ('cold ironing'). A thorough analysis requires data on the arrival and departure patterns of vessels visiting a port and also the length of time each spends at bay. Finally, for the examination of pollutants in the case the vessel cold irons it is necessary to know the energy mixture that provides power to the vessel.

There has been extensive work modeling specific nautical engines during operation and linking their emissions with the engine's load (Trozzi and Vaccaro, 1998). While there have been studies on the emissions for an entire trip, relatively little attention has been paid to the immediate area around the port as vessels arrive and depart. This local approach would help to identify the environmental 'burden' nearby residents would have due to the port's presence, while also more closely aligning with the interventions an individual port operator might make.This paper explores the potential of two different mitigation methods; Slow steaming in the immediate area of a port, and cold ironing for the hotelling activities of vessels while at berth.

Slow steaming refers to the reduction of travelling speed of vessels and has become possible due to recent increases in available shipping capacity (excess fleet).Vessels used to travel on full throttle but have been able to reduce their speed by around 10% (Psaraftis and Kontovas, 2010). While the main driver behind the introduction ofslow steaming was economical due to increasing fuel prices, indirectly there are environmental benefits. For example, Carriou (2010) shows that slow steaming manages to reduce emissions by around 11% despite the additional trips required to meet with demand. Regarding ports, similar concepts are under consideration for the last 12 or 20 nautical miles of a vessel's journey as this could greatly improve local air quality(POLA, 2009).

Cold ironing or Alternative Maritime Power (AMP) is the process of providing electrical power to a berthing ship using shore power while the ship's main and auxiliary engines are switched off. The benefits are predominantly local, as the ships' funnels do not release pollutants in port, but on the contrary the energy demand of the ship is met by the power plant (or other source of power) that feeds the port. As a result, with the use of cold ironing there are no ship emissions (at berth), although there may still be emissions elsewhere (since the electrical power supplier may still emit pollutants). To identify the induced emissions from berthing ships that cold iron, one needs to know the energy mixture the port relies on. Power provision in ports varies in regional or international level. While some ports may have (or plan to built) their own plants for their energy needs, most are still relying on power provision from the regional or national electric power industry. This means that in order to estimate induced emissions of pollutants from a port's operation, it is vital to identify where this power comes from (both location and type).

2. METHODOLOGY

2.1 Ship Arrival

The arrival pattern of ships can be approximated with a Poisson distribution (El-Naggar, 2010). The probability P(x) of the arrival of x ships in a port in a given time is expressed by

$$P(x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

where e is the base of the natural logarithm, λ is the mean number of ship arrivals in the given interval, x! is the factorial of the ship number. The departure pattern of ships depends on the total time spent at berth.This paper examines the environmental repercussions if ports were to introduce a regulation where vessels have to slow down their travelling speed for the last/first 12n.miles (buffer area) of their journey. The Port of Los Angeles has a similar scheme running, where it rewards vessel operators that comply by reducing speed to 12 knots or less in an area 20 or 40 n. miles away from the port (POLA, 2009).

If buffer area-slow steaming is universal, then no additional queues will be generated as the time interval between two successive arrivals will remain the same. This assumption normally would not hold as it is not in the interest of the port to obligate each operator to comply, so for a more precise analysis, it is necessary to have an indication of what percentage of trips would agree to reduce speed. Using these data, a stochastic analysis could explore whether queuing would arise. It should also be noted that the voyaging speed of vessels varies between 12 and 25 knots. If the buffer area would be introduced near a port where the maximum speed could not surpass 12 knots (obligatory and not voluntary policy), that would smooth vessels' traffic.

2.2 Port of Felixstowe

The port of Felixstowe is the busiest container port in the UK,situatedon the Suffolk coast, on England's eastern shores. The port is regarded as one of UK's more environmentally friendly and managed to reduce during 2010 its relative (in-port) carbon emissions by 7.2% through efficiency improvements. However the actual emissions increased by 2.1% in absolute terms (Port of Felixstowe Environmental Report 2010) due to simultaneous increases in activity. The port is less than 100 kilometers northeast of London providing good rail links to the mainland and is UK's closest port to Rotterdam (Europe's largest port). According to data from Containerisation-International, cross-referencing with Port of Felixstowe's sailing schedule, the majority of direct trips are linked with Port of Rotterdam and Port of Antwerp as shown in Table 1.

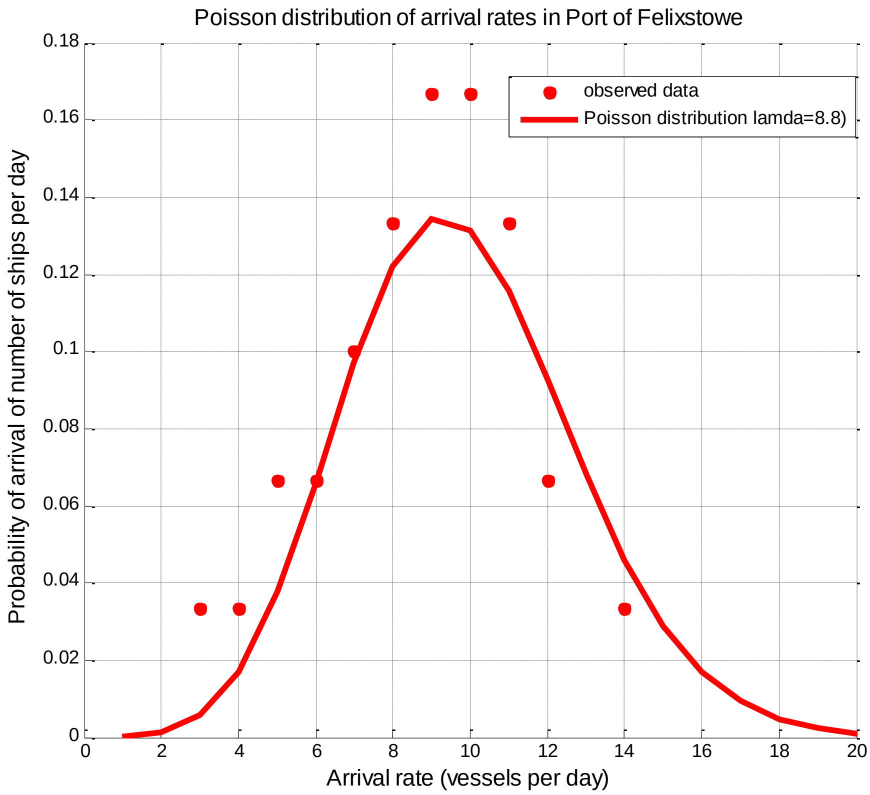


Figure 1. Distribution of probability of arrivals, port of Felixstowe

The Port of Felixstowe website provides data of scheduled arrivals and recent departures of vessels. Coupled with data from marine traffic agencies, it is possible to differentiate data from different ship sizes. Based on data from the port, in April 2012 264 vessels arrived in totalat Port of Felixstowe while 260 departed. Eurostat provides quarterly data of container volume handled from years 2007 to 2010 for major ports in the EU. According to these data there is only a small variance in volume handled at Felixstowe, with peak values independent of quarter.Using a Poisson distribution with a mean value of 8.8 vessel arrivals per day, the probability of arrivals per day is shown in figure 1 along with the observed data for April 2012.

The minimum distance for safety reasons between vessels is assumed to be 1482 meters as practiced in the Bosphorus straits (Mavrakis and Kontinakis, 2007). If ships were to reduce their speed universally to 12 knotsthere should be a difference of at least 4 minutesbetween 2 ships. On this basis, the probability of two or more successive arrivals at an interval of 4 minutes is considered negligible.

The minimum distance for safety reasons between vessels is assumed to be 1482 meters as practiced in the Bosphorus straits (Mavrakis and Kontinakis, 2007). If ships were to reduce their speed universally to 12 knotsthere should be a difference of at least 4 minutes between 2 ships. On this basis, the probability of two or more successive arrivals at an interval of 4 minutes is considered negligible.

2.3 Selection of buffer zone

The study aims to compare the potential of slow steaming near the port against the potential of cold ironing in port in terms of emissions savings. The first step to estimate the reduced emissions from slow steaming is to define an area within which ships would be asked to travel at a lower speed. As previously mentioned, POLA operates this scheme for two zones at 20n.miles and at 40n.miles offering a differential financial incentive to complying ships. For Felixstowe, selecting 20 or 40 n.miles would seem rather unrealistic for trips that in total have a length of only 200 to 400n.miles. While there are some regular direct services to distant ports (for instance Piraeus, Newark and Singapore) where a larger buffer zone would not increase significantly the total travel time (proportionally), the vast majority are shorter trips. Figure 2 presents a satellite image of port of Felixstowe with three different radii.

Given the geography of Felixstowe and the fact that for the majority of trips the next and previous ports are relatively close (see table 1), it is assumed that the port will define its buffer zone at 12n.miles which coincides with the maritime borders of the country.

It is assumed that the vessel will reduce its speed for the outer 11.5n.miles of the buffer area, as for the inner 0.5 mile the vessel would have to enter the port and start maneuvering (with extra engine load) regardless of its previous speed.

Figure 2. Satellite image of Port of Felixstowe and the area of 12,20 and 40 nautical miles (blue,red and green respectively) source: Google Earth

Table 1. Direct Distances of ports and port of Felixstowe (source: Google Earth)

Trip to/from	*Approximate Percentage of trips*	*Distance one way (nautical miles)*	*Buffer area as a % of trip*		
			12	20	40
Rotterdam	38.1%	120	10.00	16.67	33.33
Antwerp	10.3%	136	8.82	14.71	29.41
Bremerhaven	7.1%	327	3.67	6.12	12.23
Hamburg	7.9%	381	3.15	5.25	10.50
Piraeus	1.6%	2827	0.42	0.71	1.41
Newark	0.8%	3195	0.38	0.63	1.25
Singapore	2.4%	8184	0.15	0.24	0.49

2.4 Methodological approach

This paper does not deal with maneuvering emissions as it is considered that these are not altered if the vessel slowed down at the buffer area (perhaps there would be a small difference due to different engine temperature but for the purposes of this study it is neglected due to the small distances involved). This study focuses on emissions of CO_2, SO2and NO_x. Carbon and sulphur emissions depend on the amount of fuel burned, while NO_x is also affected by the engine operating speed. It is assumed that during arrival and depart time, only the main engines are operating, and while at berth only the auxiliary engines (IMO, 2008).

The fuel consumption ME_{buf}(tons of fuel) of the main engine in the buffer area is given by;

$$ME_{buf} = 10^{-6} * SFOC * EL * EP_{main} * t_{buf} \tag{1}$$

where

$SFOC$ (g/kWh) is the specific fuel oil consumption
EL is the Engine Load (%)
EP_{main}(kW) is the nominal main Engine Power installed in the vessel
t_{buf}(hours) is the time required to travel the buffer area.

To calculate the carbon emissions one has to multiply ME_{buf} with a factor of 3.17 (empirical mean value). For SO_2 according to the US EPA the formula used is;

$$E(sulphur)\left(\frac{g}{kW*hr}\right) = \alpha * FSFl + b \tag{2}$$

where α, b are unitless coefficients. α=2,3735,bis statistically insignificant (Dolphin and Melcer, 2009). $FSFl$ is the 'Fuel Sulphur Flow' ($\frac{g}{kW*hr}$) and is found by;

$$FSFl = \%S * Fuel\ Consumption \tag{3}$$

where %S is the percentage of sulphur content in fuel times fuel consumption.As Port of Felixstowe lies on the eastern shoreline of the UK, it is within the North Sea's SO_2 Emission Control Area (SOCA) as defined by Regulation 14 (Annex VI of the MARPOL convention, 2008) and thus ships are obligated to burn fuel with no more than 1 % of sulphur content until 1st July 2015 (then onwards 0.1%).

NO_xemissions depend on the engine type and the operating speed. An empirical method to calculate NO_x is by multiplying fuel consumption with appropriate emissions factor ($Nfac$)according to engine type (Kontovas and Psaraftis, 2009). $Nfac$ ranges from 0.087 for slow speed engines to 0.057 for medium speed engines (EMEP/CORINAIR 2002).
Thus the equations that will be used in this analysis are:

$$CO_2 = 3.17 * \ SFOC * EL * EP_{main} * t_{buf} \tag{4}$$
$$SO_2 = 2.3735 * 1\% * \ SFOC * EL * EP_{main} * t_{buf} \tag{5}$$
$$NO_x = Nfac * \ SFOC * EL * EP_{main} * t_{buf} \tag{6}$$

The above equations provide quantities of emissions in grams of pollutant.
In this analysis the potential for reduction of these emissions from slow steaming during arrival and departure time will be estimated. We use the accepted simplification that an increase in speed is proportional to a cubic increase in the load of the engine.

$$\frac{V_1}{V_2} = (\frac{EL_1}{EL_2})^3 \tag{7}$$

According to Carriou (2010), SFOC remains approximately unchanged for lowering speed by up to 10% but for reductions of 30% the SFOC increases (as the engine is no longer working at its optimal operational speed).Nevertheless,slow steaming is rewarding since despite the increase of SFOC the engine load drops significantly and thus overall $ME_{buf}$is reduced. Conservative assumptions are made that for a speed reduction of 30% SFOC increases by 5% and for a speed reduction of 40% SFOC increases by 10%.. Here, we use these data to examine the influence of a buffer-zone slow steaming regulation on container vessels of three different sizes and different average operating speeds that visited port of Felixstowe during April 2012. Marine traffic provided data for average operating speeds and deadweight tonnage (dwt). Data for engine power were not found and so to estimate it, equation (8) was used (adapted from EPA, 2000).

$$EP_{main}(kW) = 0.74 * (2581 + 0.719 * Dtw) \qquad)$$

The vessels' specifications are presented in table 2.

Table 2. Specifications of selected vessels (source: marinetraffic.com)

Vessel	Operating speed	Dwt	Main Engine EP(kW)	Auxiliary engine EP (kW)
A	14	8407	6384	800
B	17	118888	65166	7000

C	20	146161	79676	8000

The results are presented in table assuming that each ship travels at 12 knots for 11.5 n. miles for both arrival and departure (total distance 23 n. miles). The values for SFOC were selected according to IMO (2000).

To calculate the emissions of berthing ships the necessary elements are the time spent at berth, the engine type used (auxiliary or in rare cases main), fuel used and engine's load. The time each vessel spends at berth is easily found if the exact arrival and departure times are known. The engine's load while at berth depends on the hotelling activities of the ship, and methodologies used include empirical formulas (Kontovas and Psaraftis, 2009).Cold ironing emissions depend on the energy mixture of the country (or city) producing this power. The energy mixtureof the UK is presented in table 4 based on data from the UK Department of Energy &Climate Change (DECC).The emission factors (gr/kWh)in the UK for CO_2,SO_2,NO_x(electricity generation and grid losses) were used as 0.474, 0.3654 and 0.7566 respectively (AEA 2007). It should be noted that these provide emissions at the average level and are not to be confused with marginal emission factors which vary over time depending on current grid loading. The actual emissions generated by a berthing vessel will therefore depend on the a number of factors beyond the scope of this investigation.Average emission factors are therefore taken to provide an indicative value.

Table 3. Emissions saved at each call – slow steaming

Vessel	Average speed	%δV	%δEL	%δSFOC	%δt_{buf}	% δME_{buf}	δCO_2(kgs)	δSO_2(kgs)	δNO_x(kgs)
A	14	-14.3	-37	0.00	16.7	-26.5	1631.8	12.2	29.3
B	17	-29.4	-64.8	+0.05	41.7	-47.7	24539.6	183.7	441.2
C	20	-40.0	-78.4	+0.10	66.7	-60.4	32531.4	243.6	584.9

Table 4. Energy mix in the UK 4/2010-3/2011
(source: Department of energy & climate change)

Energy source	***%***	***Emissions at production***
Coal	28.9	✓
Natural Gas	44.2	✓
Nuclear	17.3	
Renewables	7.9	
Other	1.7	✓
Total Clean Energy	**25.2**	

The UK as member state of the European Union is obligated to increase the share of energy from renewable sources. Through the Directive 2009/28 EC, the UK has to increase the percentage to 15% by 2020. Based on this, and similar obligations for other EU Member States, cold ironing can be expected to have increasing environmental benefits on a global level. It is important to acknowledge that table one provides data for the energy generation in the UK, and does not contain information about energy imports and exports.

A critical factor for the emissions saved for cold ironing vessels is the amount spent at berth. For Port of Felixstowe in April 2012 the cumulative distribution of time is presented in figure 3a below and in figure 3b the duration against gross tonnage of vessel is shown. For this dataset the median value was 12.57 hours and the interquartile range was 13.65 hours.

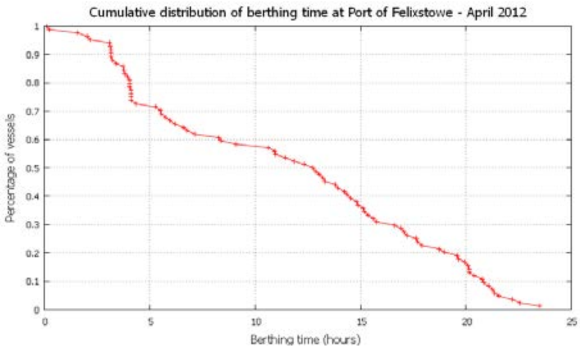


Figure 3a: Cumulative of berth duration

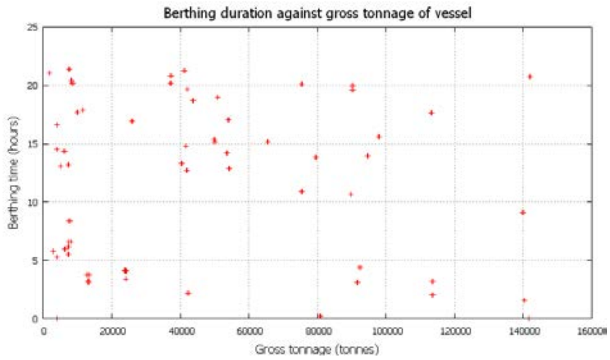


Figure 3b: Berth duration against gross tonnage

There should be an allowance of time to connect and disconnect with the power source. For the Port of Gothenburg the time needed to completethe process is less than 10 minutes (Khersonskyet *al.* 2007). Cold ironing is not intended for very short berthing durations as the vessel then would continue using its main engines.Cold start emissions are the excess emissions that occur when the engine starts below its normal operating temperature. If the berthing period is very small, cold ironing would not be as efficient as a significant portion of berthing time would be lost to plug and unplug the vessel and due to cold start emissions the environmental benefits would not be that many.

To identify the energy requirement for AMP, The ship energy demand needs to be found by dividing energy of fuel with the electric generator efficiency. This study uses a typical value of 98% as inAlkaner and Zhou (2006). It is assumed that this energy isto be provided by the grid for the case of Felixstowe. In order to estimate emissions based on that energy provided by the grid emission factors will be used.

The emissions if the vessel relies on its auxiliary engines are found using similar methodology as before through the next set of equations.

$$CO_2 = 3.17 * SFOC * EL * EP_{aux} * t_{berth} \tag{9}$$
$$SO_2 = 2.3735 * 1\% * SFOC * EL * EP_{aux} * t_{berth} \tag{10}$$
$$NO_x = Nfac * SFOC * EL * EP_{aux} * t_{berth} \tag{11}$$

where$EP_{aux}$is the nominal power of auxiliary engines installed for each ship.

It should be noted that there are examples where AMP was provided from local surplus units (for instance in Alaska the power came from hydroelectric power). It is apparent that in that case the emissions savings would vary significantly (and if RES/nuclear or hydroelectric power is used there will be no emissions at all). In Table 6 the potential of the two techniques is compared for the aforementioned calls of the three ships to Felixstowe in the UK.

Table 5. Emissions of AE versus Cold Ironing

Vessel	Berthing time(hours)	EP_{aux} (kW)	Load factor(%)	Ship energy demand(kWh)	Emissions from Auxiliary engines(kg/call)			Grid energy demand (kWh)	Emissions from grid generation(kg/call)		
					CO_2	SO_2	NO_x		CO_2	SO_2	NO_x
A	5.5	800	63	622285.7	1933.1	14.4	34.7	634985.4	301.3	1.46	0.0053
B	17	7000	63	17595000	17684.32	132.4	982.8	17954082	8518.7	13.38	0.0484
C	20.1	8000	63	2377543	20210.6	151.3	1328.1	2426064	1151.1	1.53	0.0055

Table 6. Comparison of the two techniques in emission savings (Kgr of pollutant/ call)

Vessel	Slow steaming			Cold ironing (local)			Cold ironing (global)		
	CO_2	SO_2	NO_x	CO_2	SO_2	NO_x	CO_2	SO_2	NO_x
A	1631.815	12.21802	29.34179	1933.193	14.47455	34.76088	1631.911	13.0119	34.75559
B	24539.6	183.7373	441.2483	17684.32	132.4093	982.8567	9165.647	119.0293	982.8083
C	32531.39	243.5749	584.9493	20210.65	151.3249	1328.095	8699.705	136.0335	1328.04

It is apparent that cold ironing has greater scope for longer berth durations. The local carbon effect of the two techniques is comparable for vessel A, while the larger vessels B and C seem to save more by slow steaming. Again, the time spent at berth is crucial. In other ports where the ship could dock for days it is evident that cold ironing would save more carbon emissions. Finally cold ironing is at all instances preferable regarding nitrogen and sulphur emissions as one would expect.

3. CONCLUSION

In this paper a methodology to estimate the environmental benefits of slow steaming near the port area and from cold ironing was presented. This method could be extended to represent alternative port policies for various port types. Three different vessels were examined according to real data from Port of Felixstowe and the potential of the two techniques was shown. Cold ironing performance is governed by the time spent at berth, so in order to have a comprehensive better understanding a sensitivity analysis with respects to duration of berth and auxiliary engine load factors would be suggested.Subject to data availability, the next step is to use this methodology to estimate actual emissions due to vessels in one year period in a number of ports. The scope for reduction of pollutants, in the port level, with slow steaming and cold ironing will be explored when data for each vessel's travelling speed and cold ironing 'readiness' are available. The methodology will also consider different countries with different energy mixtures, as well as a projection for the future participation of RES and nuclear (where applicable) in the mix to further explore the potential for global emissions mitigation. Finally, the two alternatives will also be evaluatedeconomically subject to data availability for cost and revenue elements where the additional delays from the slower speeds near port and the connecting/disconnecting time for cold ironing will also have to be presented in monetary terms. It is expected that the characteristics of the port size, its operational profile and surroundings will affect the extent to which each method may be implemented successfully.

ACKNOWLEDGEMENTS

This work was co-funded by the project "Scholarships Greek State Scholarship Foundation" from the resources of OP "Education and Lifelong learning" of the European Social Fund (ESF) and the NSRF 2007-2013.

REFERENCES

AEA Energy & Environment (2007), "Estimation of Rail Environmental Costs – Report to the Department for Transport" . Available at http://webarchive.nationalarchives.gov.uk/+/http:/www.dft.gov.uk/about/strategy/whitepapers/whitepapercm7176/railwhitepapersupportingdocs/nmfenvironmentalmodule.pdf [Accessed at May 2012]

Alkaner, S. & Zhou, P. (2006), A comparative study on life cycle analysis of molten carbon fuel cells and diesel engines for marine application. *Journal of Power Sources*, **158(1)**, pp.188–199.

Buhaug, *Øet al*. (2009),Second IMO Greenhouse Gas Study 2009, International Maritime Organization,

Cariou, P. (2011),'Is slow steaming a sustainable means of reducing CO2 emissions from container shipping?'.*Transportation Research Part D: Transport and Environment*,**16**, 260–264.

Defra, Department of Energy & Climate Change (2011). Guidelines to Defra/DECC's GHG Conversion Factors for Company Reporting. Available at: http://archive.defra.gov.uk/environment/business/reporting/pdf/110707-guidelines-ghg-conversion-factors.pdf [Accessed February 2012]

Dolphin, M.J. and Melcer, M. (2008). Estimation of Ship Dry Air Emissions. *Naval Engineers Journal*, **120(3)**, pp.27–36.

Environmental Protection Agency (EPA 2000), Analysis of Commercial Marine Vessels Emissions and Fuel Consumption Data. Available at: http://www.epa.gov/oms/models/nonrdmdl/c-marine/r00002.pdf [Accessed March 2012]

El-Naggar, M.E. (2010), Application of queuing theory to the container terminal at Alexandria seaport, *Journal of Soil Science and Environmental Management*, **1(4)**: 77-85

European Commission (2009). Directive 2009/29/EC amending Directive 2003/87/EC so as to improve and extend the greenhouse gas emission allowance trading scheme of the Community, 23 April 2009.

International Maritime Organization (2008). Prevention of Air pollution from ships.Available at: http://www.imo.org/blast/blastDataHelper.asp?data_id=26402&filename=INF-6.pdf [Accessed March 2012]

Khersonsky, Y., Islam, M. & Peterson, K., (2007). Challenges of Connecting Shipboard Marine Systems to Medium Voltage Shoreside Electrical Power. *Industry Applications, IEEE Transactions on*, **43(3)**, pp.838 –844.

Kontovas C.A. and Psaraftis H.N An Online Ship Emissions Calculator as a Decision-Making Aid and Policy Evaluation Tool.*13th Congress of Intl. Maritime Assoc. of Mediterranean, IMAM 2009*, İstanbul, Turkey, 12-15 Oct. 2009

Mavrakis, D., Kontinakis, N., 2008. A queueing model of maritime traffic in Bosporus Straits. *Simulation Modelling Practice and Theory*,**16**, 315–328.

Psaraftis, H.N. &Kontovas, C.A., 2010. Balancing the economic and environmental performance of maritime transportation. *Transportation Research Part D: Transport and Environment*, 15(8), pp.458–462.

Port of Felixstowe (2010)."Port of Felixstowe Environmental Report 2010". Available at: http://www.portoffelixstowe.co.uk/publications/documents/POFEnvironmentReport2010.pdf [Accessed April 2012]

Port of Los Angeles (2009)."Vessel Speed Reduction Incentive Program Guidelines". Available at: www.portoflosangeles.org/pdf/POLA VSRIP_111209_ca_cp.pdf [Accessed January 2012]

Saxe, H., Larsen, T., (2004). 'Air pollution from ships in three Danish ports'.*Atmospheric Environment***38**, 4057–4067.

Trozzi C., Vaccaro R. (1998), Methodologies for estimating air pollutant emissions from ships. Techne report MEET RF98

A VEHICLE ROUTING PROBLEM WITH CARGO TRANSFER OPTIONS

Dorota Slawa Mankowska[1], Christian Bierwirth[1] and Frank Meisel[1]

[1]*Martin-Luther-University Halle Wittenberg, Große Steinstraße 73, 06108 Halle, Germany*
dorota.mankowska@wiwi.uni-halle.de, christian.bierwirth@wiwi.uni-halle.de,
frank.meisel@wiwi.uni-halle.de

ABSTRACT

Vehicle routing problems address the routingof a vehicle fleet for servinga set of customers.In the capacitated version of the classical vehicle routing problem, vehicles are loaded only at a depot. This restricts the demand of customers visited on the route to the load capacity of the vehicle. In this paper, we investigate an extension of the problem, where one vehicle can replenish another vehicle at so-called transferring points.The replenished vehicle can continue its route and serve further customers without having to return to a depot. We describe the problem and formulate a mixed-integer programming modelfor includingcargo transfer operations into vehiclerouting. A numerical example and computational results are provided for illustrating the potentials of saving travel distance and vehicle operation timeby enabling transfer of cargo among vehicles.

Keywords:vehiclerouting, transfer of cargo, route synchronization

1 INTRODUCTION

Vehicle Routing Problems (VRPs) are intensively investigated because of their practical importance for various applications in the field of logistics and because they are difficult to solve from a computational point of view, seeGolden et al.2008. Nowadays, companies aim at achieving competitive advantage by providing increasinglycomplex logistics services, calling for the development of extended VRP models and advanced solution techniques. Such logistics services can comprise, for example, the timely visiting of customers by differently equipped vehicles or by differently skilled service operators as found in maintenance services and health care logistics. VRPs that deal withsuch complex services often involve so-called synchronization requirements for vehicle operations. This means that routes have to be designed such that two or more vehicles visit the same customer or that vehicles meet each other on their routes.An overview of routing problems with synchronization requirements is provided byDrexl (2011). These problems comprise, for example, the Split Delivery Vehicle Routing Problem wherea shipment to a customer can be shared among two or more vehicles to fulfill large demandsor to achieve a high utilization of vehicle capacity, see e. g. Archetti and Speranza (2008) and Chen et al. (2007).In the VRP with cross-docking, some vehicles collect cargo and bring it to a cross-docking terminal. At the cross-dock, the cargo is transferred to vehicles that distribute it to demand points. Clearly, the routes have to be synchronized such that the collected cargo arrives at the cross-dock before the delivery vehicles start their operations. The problem is considered for one cross-dock by Wen et al. (2008)and for multiple cross-docks by Chen et al. (2006). In the Pickup and Delivery Routing Problem with Transshipment(see Mitrovic-Minic and Laporte (2006)), trucks collect and deliver goods where cargo is consolidated to reduce distance travelled. The problem has been extended by a split-delivery option by Thangiah et al. (2007). The above problems involve cargo transfers among vehicles for increasing the vehicle utilization and for reducing the travel effort when fulfilling customers' demand. However, the locations in the logistics networks at which the cargo is transferred(depots, cross-docks etc.) are given. Direct transfer of cargo on the vehicle routes – without the need to return to a depot – has been investigated for an arc-based VRP with an application to waste collection by Del Pia and Filippi (2006). A node-based VRP,where vehicles meet on their routes for exchangingcargo (so-called truck-meets-truck traffic), has been described by Weise et al. (2009). In this problem, trucks have to meet at the same time for a direct cargo transfer. A first mathematical model for this routing problem has been presented by Mankowska et al. (2012).

In this paper, we investigate a VRP with a cargo transfer option, where one vehicle drops off cargo at transferring points that is later taken up by another vehicle. The vehicle routeshave to be synchronized such that both vehicles visit the selected transferring points in a well-defined order,i. e.the supplying vehicle drops of cargo before the receiving vehicle takes it up. Contrasting the work of Weise et al. (2009) and Mankowska et al.(2012), where vehicles have to meet directly for transferring cargo, the flexibility that is gained from allowing a temporal distance between the cargo drop-off and the take-up operation promises shorter vehicle routes. The investigation is motivated by apractical problem of DHL Germany, who sets up parcel stations (see http://www.dhl.de/packstation.html) in urban areas. The set-upof these stations is done by two heterogeneous vehicles which can load and unload parcel stations themselves. A small vehicle with a capacity of two stationscan access all set-up sites in an urban area. This vehicle is replenished by a large vehicle that carries up to twelve parcel stations but can access only some of the set-up sites. For transferring cargo, the large vehicle can leave some parcel stations at transferring points. These stations are taken up on demand by the small vehicle and brought to the set-up sites. The vehicle fleet is limited to two vehicles only, because setting-up parcel stations in a region is not a time critical process and, therefore, investments into additional vehicles are not justified from the perspective of the company.

We consider such a combined problem of vehicle routing and cargo transfer operations under a general problem setting. The task is to find vehicle routes for delivering goods to a given set of customers, where cargo transfers between vehicles are used for increasing the efficiency of vehicle operations. The cargo transfer decisions comprise the following features:

- Cargo can be transferred among vehicles at so-called transferring points. A set of potential transferring points is given. The selection of the transferring points that are actually used in a routing is an outcome of the planning process.
- The role of each vehicle, i.e. whether it provides cargo to the other vehicle or whether it receives cargo, is an outcome of the planning process. This is because the vehicle roles cannot be determined a priori. Even if vehicles show different capacities, it can be reasonable that the smaller vehicle supplies the large vehicle under certain constellations of customer time windows and restricted accessibility of customer sites.
- The scheduling of vehicles at the transferring points depends on theirassigned roles, such that the providing vehicle has to drop off the cargo before the receiving vehicle can take it up. This can effect waiting times of the receiving vehicle.

We provide a mixed-integer programming model for the considered problem together with a numerical example in the next section. Numerical experiments are presented in Section 3. Thepaper is concluded in Section 4.

2 MATHEMATICAL FORMULATION

In Section 2.1, we introduce the notation that is used for modeling the VRP with cargo transfers. The mathematical optimization model is provided in Section 2.2. A numerical example in Section 2.3 illustrates the problem and its potential outcomes.

2.1 Notation

Let C denote a set of customers that have to be served by a vehicle fleet $V = \{1,2\}$, consistingof two vehicles. The vehicles are initially located at the depot, which is represented by node 0 in the considered logistics network. The set of transferring points, at which vehicles can transfer cargo, is denoted by S. The set of all nodes in the network is denoted by $N = C \cup S \cup \{0\}$. We assume that the network is complete. The distance between any pair of nodes $i, j \in N$is given by d_{ij}. It is assumed that the travelling time from node i to node j equals the distance d_{ij}. The demand of customer $i \in C$ is denoted by q_i.The service of customer i must start within atime window $[e_i, l_i]$. A binary parameter a_{iv} is used for modeling the accessibility of customers by vehicles. It is set to $a_{iv} = 1$, if vehicle $v \in V$ can access customer $i \in C$,0if otherwise. The depot and all transferring points can be reached by both vehicles, i.e. $a_{iv} = 1$for all $i \in \{0\} \cup S$and $v \in V$. Vehicle$v \in V$has a load capacity Cap_v and requires a service duration p_{iv} for serving customer $i \in C$. Concerning transferring points, parameterp_{iv}refers to the time needed for dropping off or taking up cargo byvehicle v. For each transferring point $i \in S$, we have also given a minimal time span$\delta_i^{min}$and a maximal time span δ_i^{max} between the starting time of the drop off operation of the supplying vehicle and the starting time of the take up operation of the receiving vehicle. δ_i^{min}generally corresponds to the time needed by the supplying vehicle for dropping off the cargo. δ_i^{max}can be used for restricting the maximum time within which the cargo remains unattended at a transferring point. If both, $\delta_i^{min}$and δ_i^{max}, are set to 0, both vehicles have to start the transfer operation at the same time. This setting is applied, if the cargo must be transferred directly between the vehicles. Finally, T^{max}denotes the planning horizon of the routing problem.

The routing of the vehicles is modeled by binary decision variables x_{ijv}, which take value 1, if vehicle $v \in V$moves directly from node $i \in N$ to node $j \in N$, 0if otherwise. Continuous variables$t_{iv} \geq 0$trace the temporal progress of a vehicle route. Variable t_{iv}yields for vehicle v, the servicestart time if $i$refers to a customer, the start of a cargo transfer operation if i is a transferring point,andthe return time if $i$refers to the depot. Non-negative variables Q_{iv} represent the load of vehicle v upon arrival at node i. For each transferring point $i \in S$, the binary variableu_i determineswhether or not this point is used for a cargo transfer. To determine the role of each vehicle in a solution, we introduce binary variableα, which takes value 1, if vehicle 1 is the receiving vehicle, 0 if otherwise. Variable$\alpha$is used for ensuring a proper arrival order of the vehicles at the transferring points.

All used notation is summarized in Table 1.

Table 1: Notation.

Notation:	
$\boldsymbol{C}$	set of customers
$\boldsymbol{V} = \{\mathbf{1}, \mathbf{2}\}$	vehicle fleet
0	depot
$\boldsymbol{S}$	set of possible transferring points
$\boldsymbol{N} = \boldsymbol{C} \cup \boldsymbol{S} \cup \{\mathbf{0}\}$	set of all nodes in the network
$\boldsymbol{d_{ij}}$	distance between any pair of nodes $i, j \in N$
$\boldsymbol{q_i}$	demand of customer $i \in C$
$[\boldsymbol{e_i}, \boldsymbol{l_i}]$	time window for the service start time at customer $i \in C$
$\boldsymbol{a_{iv}} \in \{\mathbf{0}, \mathbf{1}\}$	accessibility of customers by vehicles; $a_{iv} = 1$, if vehicle $v \in V$ can access customer $i \in C$, 0 if otherwise
$\boldsymbol{Cap_v}$	load capacity of vehicle $v \in V$
$\boldsymbol{p_{iv}}$	service duration for serving customer $i \in C$; refers to the time needed for dropping off or taking up cargo by vehicle v
$\boldsymbol{\delta_i^{min}}, \boldsymbol{\delta_i^{max}}$	minimal and maximal time span between the starting time of the drop off operation of the supplying vehicle and the starting time of the take up operation of the receiving vehicle; If both, δ_i^{min} and δ_i^{max}, are set to 0, both vehicles have to start the transfer operation at the same time.
$\boldsymbol{T^{max}}$	planning horizon of the routing problem
Decision variables:	
$\boldsymbol{x_{ijv}} \in \{\mathbf{0}, \mathbf{1}\}$	takes value 1, if vehicle $v \in V$ moves directly from node $i \in N$ to node $j \in N$, 0 if otherwise
$\boldsymbol{t_{iv}} \geq \mathbf{0}$	for vehicle$v$, - the service start time if i refers to a customer, - the start of a cargo transfer operation if i is a transferring point, - and the return time if i refers to the depot
$\boldsymbol{Q_{iv}} \geq \mathbf{0}$	the load of vehicle v upon arrival at node i
$\boldsymbol{u_i} \in \{\mathbf{0}, \mathbf{1}\}$	takes value 1, if transferring point $i$is used for a cargo transfer, 0 if otherwise
$\boldsymbol{\alpha} \in \{\mathbf{0}, \mathbf{1}\}$	takes value 1, if vehicle 1 is the receiving vehicle, 0 if otherwise

2.2 Model

We consider twodifferentobjectives for the routing problem:

- Minimization of the total distance travelled by both vehicles ($\boldsymbol{Z_1}$):

$$\min \rightarrow Z_1 = \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} \cdot x_{ijv} \tag{1}$$

- Minimization of the maximal tour duration $T = \max_{v \in V}\{t_{0v}\}$ (Z_2):

$$\min \rightarrow Z_2 = T \tag{2}$$

The problem is considered under the following constraints. Constraints (3) ensure that the route of each vehicle starts and ends in the depot.

$$\sum_{j \in C} x_{0jv} = \sum_{j \in C} x_{j0v} = 1 \qquad \forall v \in V \tag{3}$$

Constraints (4) guarantee that every customer is visited by exactly one vehicle.

$$\sum_{v\in V}\sum_{j\in N} x_{ijv} = 1 \qquad \forall i \in C \tag{4}$$

Constraints (5) balances the vehicle flow by ensuring that vehicle v leaves each visited node.

$$\sum_{j\in N} x_{jiv} = \sum_{j\in N} x_{ijv} \qquad \forall i \in C \cup S, v \in V \tag{5}$$

Constraints (6) determine vehicle arrival timesat a node j. Here, M_1 refers to a sufficiently large positive value. For deriving a tight model formulation, we set $M_1 = l_i$ if $i \in C$and to $M_1 = T^{max}$if $i \ \in S$. The constraint also avoids cycles in the vehicle routes.

$$t_{iv} + \left(p_{iv} + d_{ij}\right)x_{ijv} \leq t_{jv} + M_1\left(1 - x_{ijv}\right) \quad \forall i \in N, j \in N, v \in V \tag{6}$$

Constraints (7) ensure that service operations at customers' start within their time windows.

$$e_i \sum_{j\in N} x_{ijv} \leq t_{iv} \leq l_i \sum_{j\in N} x_{ijv} \qquad \forall i \in C, \forall v \in V \tag{7}$$

Constraints (8) model the change of vehiclev's load when serving a customer i. Here, $M_2$is set to Cap_v.

$$Q_{iv} - q_i \geq Q_{jv} - M_2\left(1 - x_{ijv}\right) \qquad \forall i \in C, j \in N, v \in V \tag{8}$$

Transfers of cargo at transferring points aremodeled by Constraints (9)-(17). Constraints (9) ensure that both vehicles visit a transferring point $i \in S$only if this point is selected for beingused in a routing ($u_i = 1$).

$$\sum_{j\in N} x_{ijv} = u_i \qquad \forall i \in S, v \in V \tag{9}$$

Constraints (10) regulate the cargo flow at a transferring point $i \in S$. If node i is used and if the routing is such that vehicle 1 moves from i to node $j$and vehicle 2 moves from$i$to node$k$ (i.e. $u_i = x_{ij1} = x_{ik2} = 1$), the total load of vehicles upon arrival at nodes j and k cannot exceed the load of the vehicles upon arrival at the transferring point i. $M_3$is set to $Cap_1 + Cap_2$.

$$Q_{i1} + Q_{i2} \geq Q_{j1} + Q_{k2} - M_3\left(3 - u_i - x_{ij1} - x_{ik2}\right) \quad \forall i \in S, j, k \in N \tag{10}$$

Constraint (11)determines whether vehicle 1 is the supplying or the receiving vehicle in a solution. For this purpose, we measure if the demand of customers on vehicle 1's route exceedsitscapacity.If yes, $\alpha$is set to 1indicating that vehicle 1 is the receiving vehicle whereas vehicle 2 is the supplying vehicle. Otherwise, if the demand of customers on vehicle 1's route does not exceed its capacity, the roles of the two vehicles are swapped($\alpha = 0$).$M_4$is set toCap_2 in (11).

$$\sum_{i\in N}\sum_{j\in N} q_i x_{ij1} \leq Cap_1 + M_4 \cdot \alpha \tag{11}$$

Constraints (12)-(15) model the relationship between the start times of vehicle operations at the transferring points.If a transferring point$i$ is used ($u_i = 1$) and if vehicle 1 provides cargo for vehicle 2 ($\alpha = 0$), then the loading operation of vehicle 2 can start earliest δ_{i12}^{min}time units after the start of the drop-off operation of vehicle 1, see Constraint (12). The maximum temporal distance between the start times of the drop-off operation of vehicle 1 and the take-up operation of vehicle 2 is restricted in (13)to δ_{i12}^{max} time units. Here, M_5 is set to T^{max}.

$$t_{i2} \geq t_{i1} + \delta_{i12}^{min} - M_5(1 + \alpha - u_i) \qquad \forall i \in S \tag{12}$$

$$t_{i2} \leq t_{i1} + \delta_{i12}^{max} + M_5(1 + \alpha - u_i) \qquad \forall i \in S \tag{13}$$

The corresponding constraints for opposite roles of the vehicles are given in (14) and (15). If vehicle 1receives cargo from vehicle2 ($\alpha = 1$), then the start time of vehicle 1 starts at least δ_{i21}^{min} time units and at most δ_{i21}^{max} time unitsafter the start of the drop-off operation of vehicle 2.

$$t_{i1} \geq t_{i2} + \delta_{i21}^{min} - M_5(2 - \alpha - u_i) \qquad \forall i \in S \tag{14}$$

$$t_{i1} \leq t_{i2} + \delta_{i21}^{max} + M_5(2 - \alpha - u_i) \qquad \forall i \in S \tag{15}$$

Constraints (16) determine the value of thetotal waiting time of each vehicle for evaluating the objective function (2).

$$W_v = t_{0v} - \sum_{i \in N} \sum_{j \in N} \left(p_{iv} + d_{ij}\right) x_{ijv} \qquad \forall v \in V \tag{16}$$

The domains of the decision variables are represented by Constraints (17)-(20). Constraints (17) avoid travellingof vehicle $v$on edge(i, j), if at least one of these nodes is inaccessible for the vehicle, i. e. if$a_{iv} = 0$and/or $a_{jv} = 0$.

$$x_{ijv} \in \left\{0, a_{iv} \cdot a_{jv}\right\} \qquad \forall i, j \in N, \forall v \in V \tag{17}$$

$$t_{iv} \geq 0 \qquad \forall i \in N, \forall v \in V \tag{18}$$

$$Cap_v \geq Q_{iv} \geq 0 \qquad \forall i \in N, v \in V \tag{19}$$

$$\alpha, u_i \in \{0, 1\} \qquad \forall i \in S \tag{20}$$

2.3 Example

We consider a network with a depot 0, nine customers $C = \{1, \ldots, 9\}$, and four potential points $S = \{10, \ldots, 13\}$ for transferring cargo. Table 2 shows the instance data,which includesnode locations (coordinates x, y), customer demand, time windows, accessibility and processing times. The network is complete and the distances and travelling times between all nodes are Euclidean. The capacities of the two vehicles are set to $Cap_1 = 10$ and $Cap_2 = 50$ units. We consider here the minimization of the total distance travelled by both vehicles (Z_1). The instance is solved using ILOG CPLEX 12.3.

Table 2. Instance data for the numerical example.

	x	y	q	e	l	a_{i1}	a_{i2}	p_{i1}	p_{i2}
Depot									
0	35	38	-	0	1000	1	1	0	0
Customers C									
1	47	13	6	95	265	1	0	4	5
2	4	4	9	78	248	1	1	2	4
3	34	6	10	7	177	1	1	3	5
4	28	48	4	113	283	1	0	3	5
5	19	23	5	64	234	1	1	5	5
6	17	13	5	24	194	1	0	2	2
7	12	34	5	124	294	1	1	2	4
8	42	12	7	126	296	1	0	1	5
9	46	39	9	48	218	1	1	3	1
Transferring points S									
10	26	32	-	0	1000	1	1	1	6
11	10	49	-	0	1000	1	1	3	3
12	24	9	-	0	1000	1	1	5	6
13	0	0	-	0	1000	1	1	3	6

The optimal solution of the example instance is shown in the space-time diagram inFigure 1. Corresponding vehicle cargo values and start times of delivery and transfer operationsare provided in Table 3, where operations at transferring points are emphasized in bold. In the solution, vehicle 1 serves customers 1, 4, 5, 6, and 8 whereas vehicle 2 serves customers 2, 3, 7, and 9. The vehicles transfer cargo at the transferring points 10 and 12. Transferring points 11 and 13 are not used in the optimal routing. In this solution, vehicle 2 drops 7 cargo units at node 12 and 10 cargo units at node 10. This cargo is taken up by vehicle 1, for distributing it to the customers. In both cases, vehicle 2 arrivesat the transferring point and leaves cargo that is later picked up by vehicle 1. In other words, vehicle 2 takes the role of the cargo provider and leavescargo at transferring points without having to wait for vehicle 1. The total distance travelled by both vehicles is 288 distance units.

Table 3. Optimal vehicle routes for the example instance.

Vehicle 1	Route:	0	8	**12**	6	5	**10**	1	4	0
	t_{i1}:	-	126	**145**	158	171	**187**	216	260	275
	Cargo Q_{i1}:	-	10	**3 (takes up 7)**	10	5	**0 (takes up 10)**	10	4	0
Vehicle 2	Route:	0	9	3	**12**	2	7	**10**	0	
	t_{i2}:	-	48	84	**100**	126	161	**179**	196	
	Cargo Q_{i2}:	-	50	41	**31 (drops off 7)**	24	15	**10 (drops off 10)**	0	

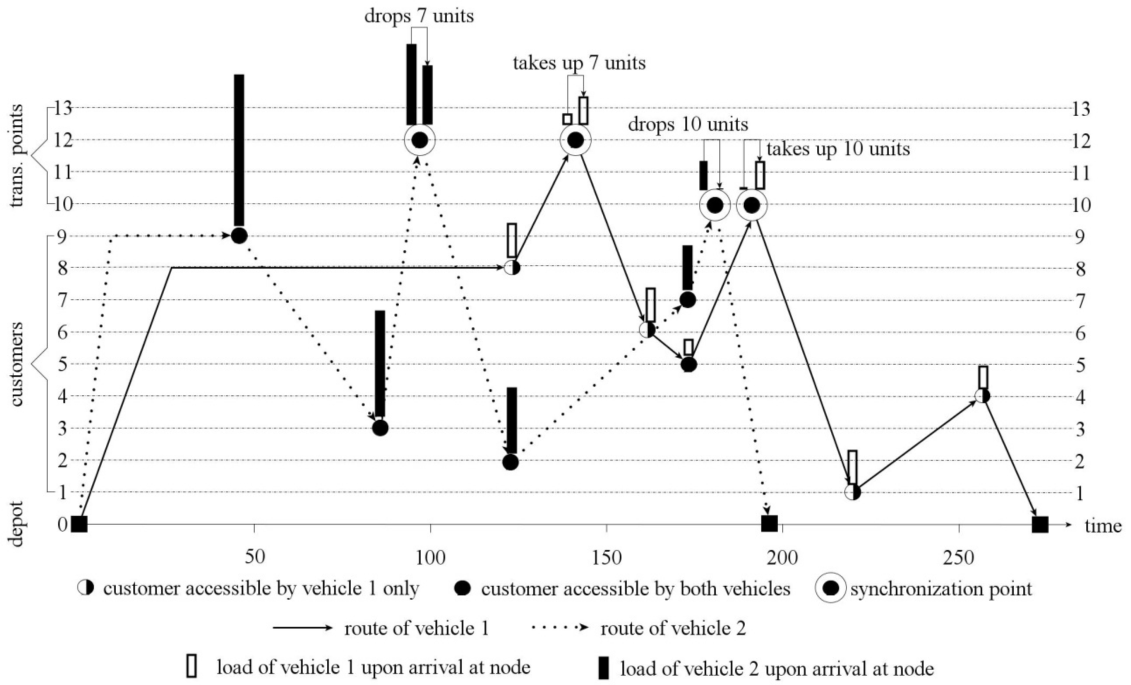


Figure 1. Time-space diagram of vehicle routes undertemporal flexibility for cargo transfers.

The provided model coversas a special case the more restrictive problem of Weise et al. (2009) and Mankowska et al.(2012), where vehicles have to meet for transferring cargo directly. This problem is obtained from the model of Section 2by setting $\delta_i^{min} = \delta_i^{max} = 0$, which enforces simultaneous vehicle operations at transferring points.The optimal solutionto this problem is shown in Figure 2. It illustrates that vehicles meet at the transferpoints 10 and 12 for exchanging cargo directly. Thesemeetings force vehicle 1 to wait until vehicle 2 arrives. The waiting, in turn, requires that the customers are visited in a different order for meeting the time windows. As a result, total distance travelled by vehicles is 297distance unitswhich is about 3% more compared with the solution of Figure 1.

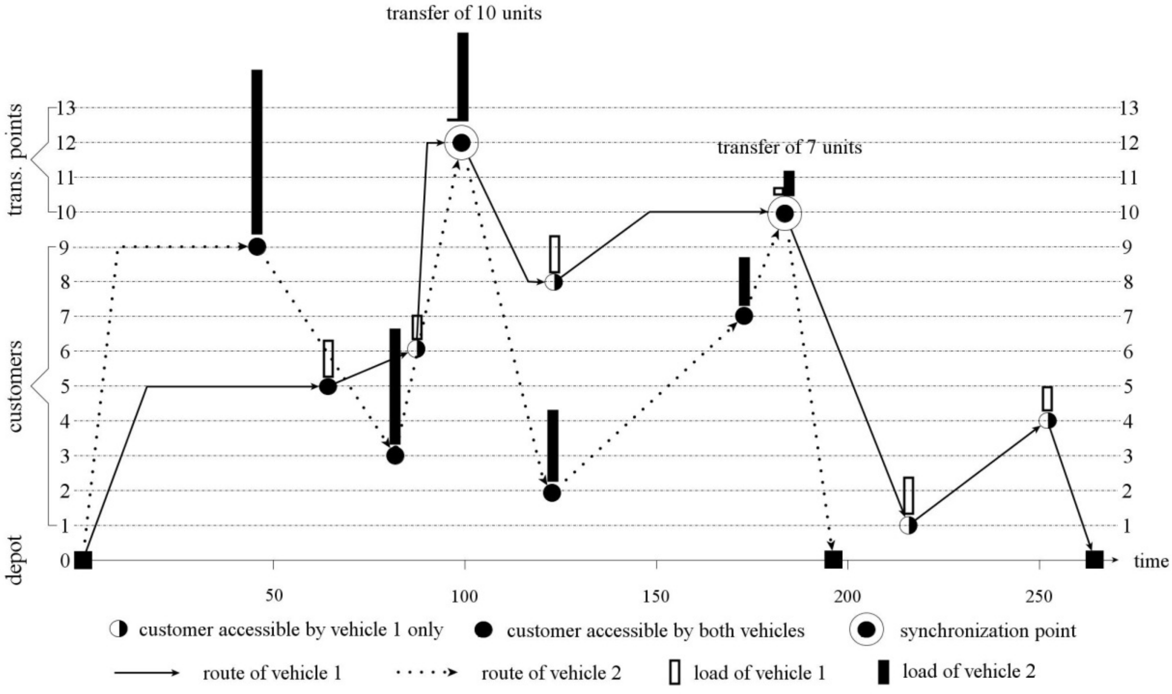


Figure 2. Time-space diagram of vehicle routes underdirect cargo transfers.

The example illustrates that flexible transfer operations, where the supplying vehicle can leave a transferring point without having to wait for the receiving vehicle, can lead to a reduction in vehicle waiting times and travel effort.

3 NUMERICAL EXPERIMENTS

We conduct numerical experiments for testing the solvability of the presented model and for conducting a sensitivity analysis regarding the different objective functions and varied vehicle capacities. We use an instance set that contains ten randomly generated test instances. Each instance consists of 12 nodes: a depot (node 0), nine customers $\boldsymbol{C} = \{\mathbf{1}, \dots, \mathbf{9}\}$, and two transferringpoints $\boldsymbol{S} = \{\mathbf{10}, \mathbf{11}\}$. All nodes are randomly placed in an area $\mathbf{50} \times \mathbf{50}$. The two vehicles are initially located at the depot. Demand of each customer is randomly drawn from an interval $[\mathbf{1}, \dots, \mathbf{10}]$. Vehicle capacities are set to $\boldsymbol{Cap_1} = \mathbf{20}$and $\boldsymbol{Cap_2} = \mathbf{40}$. The distances and travelling times are Euclidean. Service durations $\boldsymbol{p_{iv}}$ are randomly drawn integers from the interval $[\mathbf{1}, \dots, \mathbf{6}]$. For each customer $\boldsymbol{i} \in \boldsymbol{C}$, a time window of length 160 is given. The begin of a time window is randomly drawn from the interval $[\mathbf{0}, \dots, \mathbf{130}]$. The accessibility of the customers (binary parameters $\boldsymbol{a_{iv}}$) is randomly drawn such that each customer can be served by at least one vehicle. We use ILOG CPLEX 12.3 on a 3.70 GHz Intel Core i7-2600 for the computations.

The first test investigates how well the CPLEX solver can cope with the routing problem under different objectives and how intensively cargo transfers are used in a solution. Table 4 shows for each instance and pursued objective, the value of the two performance measures $\boldsymbol{Z_1}$and $\boldsymbol{Z_2}$, the total number of used transferringpoints (#u), and the computation time of CPLEX (measured in seconds). We report both performance measures to illustrate the conflict among the objective of minimizing the travelled distance and minimizing the maximum route duration. Regarding computation time, it strikes that the minimization of the total distance travelled by both vehicles ($\boldsymbol{Z_1}$) is comparably easy for CPLEX, whereas the maximal tour duration ($\boldsymbol{Z_2}$) appears to be more difficult. Regarding columns #u, it is found that the average number of transfers islower if the maximal tour durationis minimized ($\boldsymbol{Z_2}$) instead of minimizing the total distance travelled ($\boldsymbol{Z_1}$). When pursuing $\boldsymbol{Z_2}$, cargo transfers are included in route only if it is indispensableforserving customers, whereas under $\boldsymbol{Z_1}$ additional transfers are included just for shortening the travelled distance. This illustrates that cargo transfers are not necessarily mandatory in a routing.They can also be motivated by improvements of the solution quality. In other words, enabling cargo transfers in a VRP opens up potential for finding solutionsof higher quality.

Table 4. Numerical results for two different objective functions.

	$\boldsymbol{Z_1}$ pursued				$\boldsymbol{Z_2}$ pursued			
No.	$\boldsymbol{Z_1}$	$\boldsymbol{Z_2}$	#u	CPU	$\boldsymbol{Z_1}$	$\boldsymbol{Z_2}$	#u	CPU
1	230.5	231.0	1	≤1	255.3	189.6	0	8
2	250.4	223.0	1	≤1	263.8	170.5	1	2
3	201.4	166.0	0	≤1	239.7	154.0	0	2
4	188.9	195.0	0	≤1	204.9	160.0	0	15
5	274.5	214.0	1	≤1	295.6	191.0	1	1
6	215.5	211.0	1	≤1	226.0	160.4	1	8
7	243.4	225.0	1	≤1	250.5	173.5	0	4
8	222.0	160.0	1	≤1	222.9	159.3	1	3
9	167.6	242.0	1	≤1	207.1	166.0	1	6
10	203.1	172.0	0	≤1	211.8	163.6	0	1
∅	219.7	203.9	0.7	≤1	235.8	168.8	0.5	5

In the second test, we vary the capacity of the vehicles. We consider vehicle capacities $(\boldsymbol{Cap_1}, \boldsymbol{Cap_2}) = (\mathbf{10}, \mathbf{50})$, $(\mathbf{20}, \mathbf{40})$ and $(\mathbf{30}, \mathbf{30})$. Table 5 shows the computational results for these three settings. For one instance (instance 5 under setting $\boldsymbol{Cap_1} = \mathbf{10}$ and $\boldsymbol{Cap_2} = \mathbf{50}$) no feasible solution is found. Here, the strongly differing vehicle capacities necessitate that vehicle 1 is replenished three times by vehicle 2 in order to serve those customers that are accessible by vehicle 1 only. However, only two transferring points are available in this instance. Since multiple visits of one transferring point is not supported by the optimization model, no feasible routing can be obtained here. For all other instances, optimal solutions are found within less than 1 second. As expected, the number of cargo transfers (columns #u) is larger if the vehicles differ strongly in their capacities. The transfer operations cause additional travelling to the transferring points, which leads to an average total travelled distance of 265.3 under capacities (10,50). If vehicle capacities differ by only 20 units (20,40) or even if they are the same (30,30) the number of transfer operations and the total travelled distances decrease clearly. Nevertheless, it is most interesting to see that cargo transfers still occur in the optimal routings even if vehicles show the same capacity $\boldsymbol{Cap_1} = \boldsymbol{Cap_2} = \mathbf{30}$. This illustrates that not just heterogeneous vehicle capacities call for cargo transfers in a routing problem. Also a restricted accessibility of customers or time windows for the services can necessitate cargo transfer operations. The variety of reasons for transferring cargo among vehicles illustrates the importance of including these operations into vehicle routing problems.

Table 5. Numerical results for different vehicle capacities (Cap_1, Cap_2).

	(10,50)			(20,40)			(30,30)		
No.	Z_1	#u	CPU	Z_1	#u	CPU	Z_1	#u	CPU
1	266.4	1	≤1	230.5	1	≤1	235.3	1	≤1
2	355.8	2	≤1	250.4	1	≤1	218.5	0	≤1
3	238.3	1	≤1	201.4	0	≤1	208.8	0	≤1
4	215.2	1	≤1	188.9	0	≤1	178.5	0	≤1
5	-	-	≤1	274.5	1	≤1	260.2	1	≤1
6	227.4	1	≤1	215.5	1	≤1	218.2	1	≤1
7	250.8	1	≤1	243.4	1	≤1	241.9	1	≤1
8	273.7	2	≤1	222.0	1	≤1	201.0	0	≤1
9	313.9	2	≤1	167.6	1	≤1	165.3	1	≤1
10	246.1	1	≤1	203.1	0	≤1	203.5	0	≤1
∅	265.3	1.3	≤1	219.7	0.7	≤1	213.1	0.5	≤1

4 CONCLUSION

In this paper, we present a vehicle routing problem where cargo can be transferred among vehicles. The presented model supports planning of operations for two vehicles, where cargo transfer can occur directly or with some flexibility in the scheduling of drop-off and take-up operations. A numerical example illustrates the advantage that is gained from a temporal flexibility in the transfer operations. A limitation of the approach has been revealed in the computational study. Since the model does support at most one cargo transfer at each transferring point, instances with strongly differing vehicle capacities can probably not be solved feasibly. This finding motivates to generalize the approach towards an arbitrary number of visits at a transferring point and also towards large vehicle fleets.

ACKNOWLEDGEMENTS

This research is funded by the German Research Foundation (DFG) under project reference B02110263.

REFERENCES:

Archetti, C. and Speranza, M. (2008), The Split Delivery Vehicle Routing Problem: A Survey, In B. Golden, S. Raghavan and E. Wasil (eds): *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, New York, 103-122

Bredström, D. and Rönnqvist, M. (2008), Combined vehicle routing and scheduling with temporal precedence and synchronization constraints, *European Journal of Operational Research*, **191**, 19-31

Chen, S., Golden, B. and Wasil, E. (2007), The Split Delivery Vehicle Routing Problem: Applications, Algorithms, Test Problems and Computational Results, *Networks*, **49**, 318-329

Chen, P., Guo, Y., Lim, A. and Rodrigues, B. (2006), Multiplecrossdocks with inventory and time windows, *Computers & Operations Research*, **33**, 43-63

Del Pia, A. and Filippi, C. (2006), A variable neighborhood descent algorithm for a real waste collection problem with mobile depots, *International Transactions in Operational Research*, **13**, 125-141

Drexl, M. (2011), Synchronization in Vehicle Routing – A Survey of VRPs with Multiple Synchronization Constraints, *Transportation Science*, 1-20, in press

Golden, B., Raghavan, S. and Wasil, E. (eds) (2008), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, New York

Mankowska, D., Bierwirth, C. and Meisel, F. (2012), Spatial and temporal synchronization of vehicles in logistics networks, in Klatte, D. et al. (eds): *Operations Research Proceedings 2011*, 1-6, in press

Mitrovic-Minic, S. and Laporte, G. (2006), The Pickup and Delivery Problem with Time Windows and Transshipment, *INFOR*, **44**, 217-227

Thangiah, S. R., Fergany, A. and Awan, S. (2007), Real-Time Split-Delivery Pickup and Delivery Time Window Problems with Transfers, *Central European Journal of Operations Research*, **15**, 329-349

Weise, T., Podlich, A., Reinhard, K., Gorldt, Chr. and Geihs, K. (2009), Evolutionary Freight Transportation Planning, In M. Giacobiniet. al (eds), *Applications of Evolutionary Computing – Proceedings of EvoWorkshop 2009*, **5484/2009**, 768-777

Wen, M., Larsen, J., Clausen, J., Cordeau, J. and Laporte, G. (2008), Vehicle Routing with Cross-Docking, *Journal of Operational Research Society*, **60**, 1708-1718

ANADAPTIVE NEIGHBORHOOD SEARCH APPROACH FOR THE VRPSPDTL

Olcay Polat[1], Osman Kulak[2] and Hans-Otto Günther[1]

[1]*Technical University of Berlin, Department of Production Management, Straße des 17.Juni 135, 10623 Berlin, Germany, opolat@pau.edu.tr, hans-otto.guenther@tu-berlin.de*

[2]*Pamukkale University, Department of Industrial Engineering, Kinikli, 20100 Denizli, Turkey, okulak@pau.edu.tr*

ABSTRACT

The Vehicle Routing Problem with Simultaneous Pickup and Delivery with Time Limit (VRPSPDTL) is a variation of the Vehicle Routing Problem where the vehicles serve delivery as well as pick up operations of the clients under time limit restrictions. The VRPSPDTL determines a fleet of vehicle routes starting and ending at a central depot by serving all clients just once under the objective of minimizing the total distance travelled. For this problem, we propose a mixed integer mathematical optimization model and a novel adaptive neighborhood search algorithm combined with the classic savings heuristic, variable neighborhood search, and a perturbation mechanism. The numerical results show that the proposed method produces superior solutions compared to those reported in the literature for a number of well-known benchmark problems and reasonably good solutions for the remaining test problems.

Keywords: vehicle routing, meta-heuristics, adaptive neighborhood search

1 INTRODUCTION

The Vehicle Routing Problem (VRP) refers to serving a set of clients from a central depot by a homogeneous fleet of capacitated vehicles. This problem aims to determine a set of vehicle routes starting and ending at the central depot by serving all clients just once under the objective of minimizing the total distance travelled. A variation of the VRP is the Vehicle Routing Problem with Pickup and Delivery (VRPPD) where the vehicles serve both delivery and pick up operations at client locations. The VRPPD can be categorized into three classes (Nagy and Salhi, 2005):

- VRP with Backhauls (VRPB): the vehicles first serve delivery operations, next pick up operations at clients.
- Mixed VRPB (MVRPB): the vehicles serve delivery or pick up operations to clients in any sequence.
- VRP with Simultaneous Pickup and Delivery (VRPSPD): the vehicles simultaneously serve delivery and pick up operations to clients

In the literature, the VRPSPD was proposed by Min (1989). In this problem, the existing load of the vehicles has to be checked at each client to guarantee that the vehicle capacity is not violated. In this study, we consider the VRPSPD with Time Limit (VRPSPDTL). The VRPSPDTL was first considered by Salhi and Nagy (1999). The authors impose a predefined service time for the clients and a maximum total duration (travel + service time) restriction for the vehicles in the VRPSPD.

Dethloff (2001) defined the VRPSPD as an NP-hard combinatorial optimization problem, which means that practical large-scale problem instances cannot be solved by exact solution methodologies within acceptable computational time. In the VRPSPDTL the objective and constraints are the same as in the VRPSPD, except the service time limit of vehicles, which makes the problem more complicated because of the difficulty due to the control of the voyage duration of the vehicle in addition to the service time of the clients along the route. As a result, this problem can be described as NP-hard, too. In the literature the interest was therefore focused on heuristic solution approaches.

Recently a number of authors proposed heuristic and meta-heuristic implementations for VRPSPD benchmark problems. However, just few of these researches considered the benchmark problems with time limit proposed by Salhi and Nagy (1999). Among these are: Cluster Insertion heuristics (CI) by Salhi and Nagy (1999), Insertion Based Heuristics (IBH) by Detholff (2001), Large Neighborhood Search (LNS) by Ropke (2006), Reactive Tabu Search (RTS) by Wassan et al. (2007), Ant Colony System (ACS) by Gajpal and Abad (2009), Saving Based Ant Algorithm (SBAA) by Catay (2010), and Nearest Sweep with Perturbation (NSP) by Jun and Kim (2011) who proposed heuristic and metaheuristic implementations for the VRPSPD as well as for the VRPSPDTL.

Subramanian and Cabral (2008) presented the first investigation dealing with the pure VRPSPDTL considering the CMT 6-7-8-9-10-13-14 X&Y benchmark problems of Salhi and Nagy (1999). The authors proposed an Iterated Local Search (ILS) procedure in order to solve the problem.

In this study, we propose a novel adaptive neighborhood search (ANS) approach for the VRPSPDTL. The remainder of this study is structured as follows. In Section 2, we present a model formulation for the problem at hand. Next, the solution procedure is developed. Detailed numerical results are presented in Section 4. Finally, conclusions are drawn and suggestions for further research are given in Section 5.

2 THE VEHICLE ROUTING PROBLEM WITH PICKUP AND DELIVERY AND TıME LıMıT

The problem considered is that of designing the network of service vehicles, i.e. simultaneously dispatching/collecting cargo parcels from a central post station to/from regional post stations via trucks, simultaneously dispatching/collecting containers from a hub port to/from feeder ports via containerships, simultaneously dispatching/collecting passengers from a continental center airport to/from national airports via airplanes, etc. In this concept, the VRPSPDTL can be stated as follows. A set of clients is located on a distribution network where clients require both delivery and pickup operations. Each client has to be served once for both operations with a given fleet of identical capacitated vehicles. Each vehicle leaves the central depot carrying the total amount of goods it has to deliver and returns to the central depot carrying the total amount of goods it must pick-up. Each client also has a specified service time which is the loading and unloading operation time of the vehicle at the client. Therefore, the voyage time of a vehicle is the sum of total travel time of the route and total service time of the clients. In order to determine the vehicle schedules and the staffing balance, each vehicle has to finish its voyage before the maximal allowed duration is reached.

A mixed integer linear programming (MILP) model for the VRPSPDTL can be formulated as follows:

Indices:

$i, j \in N$	clients
$i, j \in N_0$	clients plus depot (client 0): $N_0 = N \cup \{0\}$
$k \in K$	vehicles (K=maximum number of vehicles)

Parameters:

R	maximum allowed voyage duration of vehicles (travel time + service time)
Q	maximum load capacity of vehicle

v	average travel speed of vehicle
c_{ij}	distance between client i and j (i, j ∈ N_o)
s_i	service time at client i (i, j ∈ N)
d_i	delivery demand of client i (i, j ∈ N)
p_i	pick-up demand of client i (i, j ∈ N)

Decision variables:

$x_{jik} = \begin{cases} 1, & \text{if the arc between client i and j belongs to the route served by vehicle k;} \\ 0, & \text{otherwise.} \end{cases}$

y_{ij}	goods picked up from clients up to client i and transported from client i to j
z_{ij}	demand to be delivered to clients routed after client i and transported between client i and j

The model formulation is given as follows:

$$min \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} c_{ij} x_{ijk} \tag{1}$$

s.t.

$$\sum_{i=0}^{N_0} \sum_{k=1}^{K} x_{ijk} = 1, \qquad j \in \mathrm{N} \tag{2}$$

$$\sum_{i=0}^{\mathrm{N}_0} x_{ijk} - \sum_{i=0}^{\mathrm{N}_0} x_{jik} = 0, \qquad j \in \mathrm{N}_0; \mathrm{k} \in K \tag{3}$$

$$\sum_{j=1}^{N} x_{0jk} \leq 1, \qquad \mathrm{k} \in K \tag{4}$$

$$\sum_{i=0}^{\mathrm{N}_0} y_{ji} - \sum_{i=0}^{\mathrm{N}_0} y_{ij} = p_j, \qquad j \in \mathrm{N} \tag{5}$$

$$\sum_{i=0}^{\mathrm{N}_0} z_{ij} - \sum_{i=0}^{\mathrm{N}_0} z_{ji} = d_j, \qquad j \in \mathrm{N} \tag{6}$$

$$y_{ij} + z_{ij} \leq Q \sum_{k=1}^{K} x_{ijk}, \qquad i, j \in \mathrm{N} \tag{7}$$

$$\sum_{i=0}^{\mathrm{N}_0} \sum_{j=0}^{\mathrm{N}_0} \frac{c_{ij}}{v} x_{ijk} + \sum_{i=1}^{\mathrm{N}} \sum_{j=0}^{\mathrm{N}_0} s_i x_{ijk} \leq R, \qquad \mathrm{k} \in K \tag{8}$$

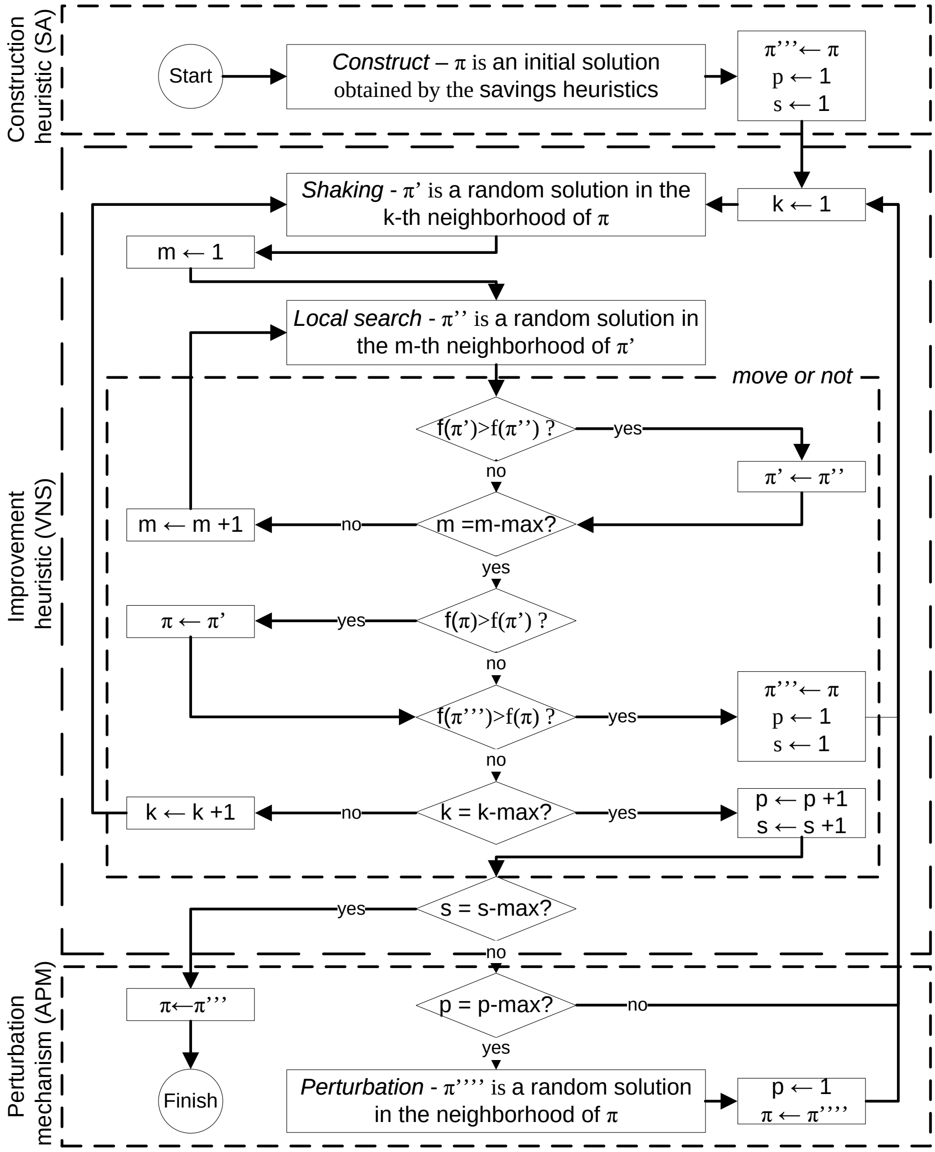
$$x_{ijk} \in \{0,1\}, y_{ij} \geq 0, z_{ij} \geq 0, \qquad i, j \in \mathrm{N}_0; \mathrm{k} \in K \tag{9}$$

The objective function (1) aims to minimize the total travelled distance. Equation (2) ensures that each client is served by only one vehicle. Equation (3) guarantees that the same vehicle arrives at and departs from each client. Restrictions (4) ensure usage of maximum K vehicles. Equations (5) and (6) satisfy pick-up and delivery demand of the clients, respectively. Restrictions (7) are the vehicle capacity constraint. Restrictions (8) represent the maximum voyage duration constraint. Finally, constraints (9) define the variable domains. In general, the constraints ensure that each vehicle departs from the central depot with a load equivalent to the total delivery goods and each vehicle arrives at the central depot with a load equivalent to the total pick-up goods from clients in the route served by that vehicle.

The performance of the presented mathematical model has been tested for various problem sizes by use of GAMS 23.7 software with the CPLEX 12 solver. The solutions turned out to be computationally demanding when problem sizes exceeded 14 clients.

3 THE PROPOSED METHODOLOGY

As mentioned in sections 1 and 2, the exact approaches suggested for solving the VRPSPTL are not practical for large-sized problem instances. In this study, we therefore propose a novel adaptive neighborhood search (ANS) algorithm based on heuristic approaches. The steps of the approach are described in Fig 1. The algorithm applies the savings algorithm (SA) in order to gain a fast and effective initial solution. The ANS is embedded with variable neighborhood search (VNS) to improve the initial solution by searching neighborhoods. In order to escape from local optima, an adaptive perturbation mechanism (APM) is developed.



[*] m-max: number of local search structures of VNS; k-max: number of shaking structures of VNS; s-max: termination number of ANS; p-max: perturbation call number of APM

Fig. 1 Structure of the ANS approach

The initial solution is constructed by using the savings algorithm of Clarke and Wright (1964). This classic heuristic aims at merging sub-tours based on costs savings which can be achieved by combining two sub-tours to be served by one vehicle. In the literature, some enhancements of the Clarke and Wright savings algorithm have been suggested by adding new terms and parameterizing the savings formula. In this study, we use the savings formula proposed for the VRP by Altinel and Öncan (2005).

Afterwards the initial solution is evaluated with an VNS improvement algorithm. The VNS, which is based on the idea of systematically changing the neighborhoods in order to improve the current situation, was introduced by Mladenović and Hansen (1997). VNS aims to explore the solution space which can not been searched by local search. *Shaking*, *local search* and *move or not* operators are used in the implementation of the VNS. The shaking operator defines the search direction of the VNS by using the set of neighborhoods. The opportunity of reaching a global solution improves when combining the shaking operator with local search rather than using a single shaking operator. Therefore, each solution obtained through the shaking operator is used in the local search operator in order to explore promising new neighborhoods of the current solution. In this study we implemented the variable neighborhood descent (VND) algorithm as the local search operator. The VND aims to combine the set of neighborhoods in a deterministic way, since using more than one neighborhood structure could obtain a better solution (Hansen and Mladenović, 2001).

In this study, a set of N_k neighborhood structures is employed in a deterministic order as *shaking and local search operators* (Fig. 2). To avoid unnecessary movements, only feasible movements are admitted, i.e. those that do not violate the ship capacity and total duration limit of the route. Also, a reversed version of the routes which violate the vehicle capacity is applied, if the total delivery and pick-up amounts of the route are feasible.

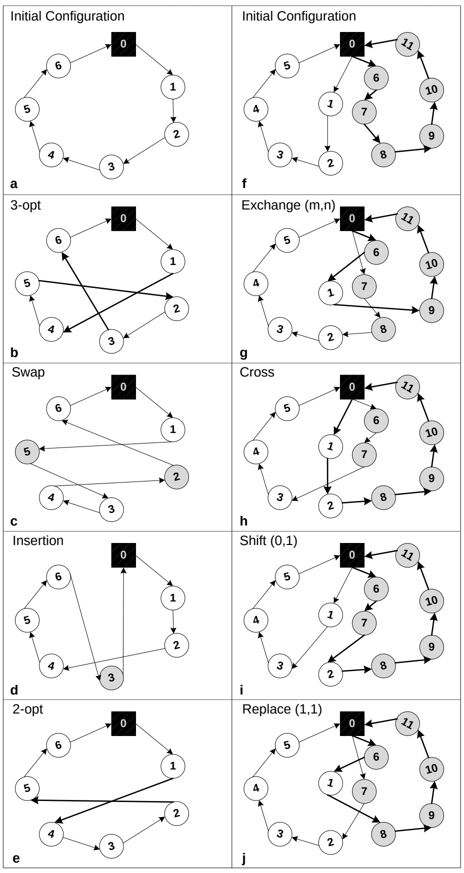


Fig. 2 Neighborhood structures

In all inter-route neighborhood structures, the route pairs are selected according to the roulette wheel principle in order to eliminate the number of unfeasible exchange operators. In route selection, after a route is randomly selected, another route is selected according to the center of gravity based on distances after they are calculated and scaled. All the distance between the selected route and remaining routes are scaled by a $1/distance^{0.5}$ factor in order to give exchange opportunity to distant routes.

The temporary solution which is obtained after the shaking and local search operators are applied is compared with the current solution in order to decide whether to *move or not.* In the proposed VNS and VND, the acceptance criterion of the temporary solution accepts only improvements. This procedure, however, could simply stick the search to a local optimum. Therefore, it is necessary to employ a strategy of accepting non-improving solutions. Perturbation is an effective strategy used to jump out of the local optimum and to search a new promising region. A commonly used perturbation strategy is to destruct the previous local optimum partially in a random way (e.g. Subramanian et. al 2010). Another strategy is the destroy-and-repair based perturbation mechanism (e.g. Jun and Kim, 2012).

The previously obtained local optimum solution combines global statistical information and local information of good individual solutions. In this study, the current solution therefore is used to develop a novel perturbation method called adaptive perturbation mechanism (APM). This perturbation mechanism runs after a number of non-improving iterations counted from the last improving solution. In the APM, a set P_x {P_1: *double replace,* P_2: *double cross,* P_3: *triple shift,* P_4: *triple replace,* P_5: *triple cross*} perturbation structures is randomly run whenever the perturbation is called. In addition to the perturbation move, a local search with previously defined four intra-route neighborhood structures is applied in order to improve the perturbed solution quality which is important since a perturbation move satisfying vehicle capacity and total route duration limit is always accepted. Moreover, violating moves are accepted only if the total route duration and vehicle capacity are below an acceptance limit (α). However, just one of the routes is allowed to use this violation and the travel duration of this route is punished with a very big penalty cost. This rule gives routes a potential improvement chance in the shaking and local search phase. The new developed perturbation structures for the APM are defined below:

Double Replace (P_1) is a combination of two times sequential *Replace (1,1)* movement to the same routes which are selected by the *roulette wheel* method. A random client from route 1 is permutated with a random client from route 2; next, another random client from route 1 is permutated with a client from route 2. After the intra local search is applied to both route1 and route 2, the total vehicle duration and vehicle load capacity are checked according to the acceptance limits.

Double Cross (P_2) applies the *Cross* exchange. Otherwise, it is similar to the *Double Replace* structure.

Triple Shift (P_3) is a newly developed fast and effective perturbation movement to jump out from local optima. A route (route 1) is randomly selected, and two another routes (route 2 and route 3) are selected by using the defined roulette *wheel* method. Next, similar to the *Shift (0,1)* movement a random client from route 2 is transferred to route 1, and a client from route 1 is transferred to route 3. Similar to double structures, vehicle duration and capacity are checked according to the acceptance limit after intra-local search applied to routes.

Triple Replace (P_4) is similar to *Triple Shift* by using the *Replace (1,1)* movement.

Triple Cross (P_5) is similar to *Triple Shift* by using the *Cross* exchange structure.

As local search, a set of L_y {L_1: *3-opt*, L_2: *Swap,* L_3: *Insertion*, L_4: *2-opt*} intra neighborhood structures are used in deterministic order. If no feasible solution is generated after z_{max} attempts in any perturbation structure, the algorithm then tries another perturbation structure.

4 NUMERICAL RESULTS

The proposed heuristic algorithm is coded using Matlab R2010b and executed on an Intel Core-2-Duo T5750 2.0 GHz processor with 3 Gb RAM. The performance of the proposed algorithm is tested using a benchmark problem set for the VRPSPDTL from Salhi and Nagy (1999) based on Christofides et al. (1979). This problem set includes 14 problem instances in which client numbers vary between 50 and 199. Salhi and Nagy (1999) manipulated 7 original VRP benchmark problem instances of Christofides et al. (1979) by imposing a maximum time restriction for the vehicles, giving a predefined service time, and splitting the original demand between pickup and delivery loads. The remaining 7 instances were obtained by switching these pickup and delivery loads.

As a part of preliminary studies, experiments on the sequence of the shaking operators of the VNS algorithm were conducted in order to determine the most effective sequence of the local neighborhood search set. The results demonstrated the effectiveness of {N_1: *3-opt*, N_2: *Swap*, N_3: *Insertion*, N_4: *2-opt*, N_5: *Exchange (m,n)*, N_6: *Cross*, N_7: *Shift (0,1)*, N_8: *Replace (1,1)*} sequence. The same sequence is used in the local search (VND) part of the VNS algorithm. Therefore, k_{max} and m_{max} parameters of the VNS algorithm set to 8 in the experiments.

The perturbation mechanism is called after $3 * current\ total\ route\ number$ (p_{max}) of iterations counted from the last accepted move. The total route duration violation acceptance limit (α_1) is used as *the service time via allowing just one client*. This rule aims to allow clients to join another route for possible future improvements. Also the vehicle capacity violation acceptance limit (α_1) is used as the maximum of all pick-up and delivery loads.

The termination condition of the ANS algorithm is used as maximum number of iterations between two improvements of the best solution. The termination condition is set to 10000 iterations without improvement. The proposed ANS algorithm is run ten times with different random seeds for each benchmark problem in order to measure its robustness.

The proposed ANS heuristic is compared with the best solutions of IBH by Detholff (2001), LNS by Ropke (2006), RTS by Wassan et al. (2007), ILS by Subramanian and Cabral (2008), SBAA by Catay (2010), and NSP by Jun and Kim (2011) for benchmark problem instances of Salhi and Nagy (1999). The detailed results of the comparison of all seven approaches are given in Table 1. The best solutions of the problem types are highlighted using bold type.

Table. 1 Computational results for the benchmark problem instances

Instance	#N*	IBH* Best	LNS* Best	RTS* Best	ILS* Best	SBAA* Best	NSP* Best	ANS Best*	ANS #Veh*	ANS Avg.S.*	ANS %Gap*	ANS B.T.
CMT6X	50	584	559	556,06	**555,43**	558,68	**555,43**	**555,43**	6	556,67	0,00	47.
CMT7X	75	961	**901**	903,05	**901,22**	**901,22**	**901,22**	**901,22**	11	901,22	0,00	70.3
CMT8X	100	928	866	879,60	**865,50**	**865,51**	**865,50**	**865,50**	9	865,50	0,00	224.6
CMT9X	150	1299	1197	1220,00	1167,23	1173,44	**1161,37**	**1161,37**	14	1162,84	0,00	484.0
CMT10X	200	1571	1462	1464,58	1407,66	1424,06	1392,36	**1390,92**	18	1391,52	**-0,10**	1168.8
CMT13X	120	1576	1578	1647,51	1545,96	1551,25	1549,79	**1542,86**	11	1543,17	**-0,20**	332.7
CMT14X	100	871	863	823,95	**821,75**	**821,75**	**821,75**	**821,75**	10	821,75	0,00	228.5
CMT6Y	50	584	559	558,17	**555,43**	556,68	**555,43**	**555,43**	6	555,43	0,00	47.3
CMT7Y	75	961	952	903,36	901,22	901,22	**901,10**	901,22	11	901,22	0,01	69.8
CMT8Y	100	936	873	917,42	**865,50**	**865,51**	**865,50**	**865,50**	9	865,50	0,00	162.7
CMT9Y	150	1299	1256	1213,11	1167,69	1171,95	**1161,37**	**1161,37**	14	1162,58	0,00	527.7
CMT10Y	200	1571	1552	1419,79	1413,88	1429,46	1392,36	**1390,92**	18	1391,95	**-0,10**	1097.4
CMT13Y	120	1576	1602	1647,04	**1542,86**	1547,75	1544,37	**1542,86**	11	1542,86	0,00	375.3
CMT14Y	100	871	n/a	823,34	**821,75**	822,35	**821,75**	**821,75**	10	821,75	0,00	204.6
Average*	-	1113	-	1069,78	1038,08	1042,20	1034,95	**1034.15**	-	**1034,57**	**-0,03**	360.1

* #N: Number of nodes; #Veh: Number of vehicles; Best: Best solution; Avg.S.: Average solution; %Gap: Percentage difference between the best known and VNS; B.T.: Best soltion finding CPU time; Average: Average of 14 instances

Among 14 instances, the ANS approach could generate a new best solution for three problem instances, namely CMT10X, CMT10Y and CMT13X. In addition, ANS reproduces best-known solutions for 10 instances. For the remaining CMT7Y instance, the gap between the results of the ANS and the best-known solution to the NSP is just around 0.01%**.** Indeed, ANS shows the lowest average solutions for the considered instances. The general conclusion that can be drawn from Table 1 is that the ANS algorithm represents adequate solutions in an acceptable solution time over the benchmark problems of Salhi and Nagy (1999).

5 CONCLUSION

In this study, we proposed a novel hybrid search method called adaptive neighborhood search (ANS) algorithm based on the savings algorithm (SA), variable neighborhood search (VNS) and adaptive perturbation mechanism (APM) to solve the vehicle routing problem with simultaneous pick-up and delivery time limit (VRPSPDTL). We used eight local neighborhood search structures as shaking and local search operators of the VNS algorithm. A variable neighborhood descent (VND) procedure is used to perform the local search. We used five adaptive perturbation structures in order to escape from local optima. From the numerical results it can be concluded that the proposed ANS algorithm generates efficient solutions compared to existing solution methods for the VRPSPDTL. For 13 out of the 14 benchmark instances from Salhi and Nagy (1999), the ANS algorithm could obtain new best solutions or reach the best known solution. The main features of the proposed ANS algorithm are specifically designed sub-procedures as part of the construction heuristic, improvement algorithm and perturbation mechanism to cover the total vehicle duration limit which is not included in pure VNPSPD solution methods. The proposed ANS algorithm can be adapted in order to consider heterogeneous fleet conditions and dynamic environment of the VRPSPDTL.

REFERENCES

Altınel, K., Öncan, T. (2005) A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *Journal of the Operational Research Society* 56, 954–961

Catay, B. (2010) A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Expert Systems with Applications* 37, 6809-6817.

Christofides, N., Mingozzi, A., Toth, P. (1979) The vehicle routing problem. *Combinatorial optimization* ed Christofides, A.M., P. Toth, & C. Sandi Wiley, Chichester, pp. 315-338.

Clarke, G., Wright, J.W. (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12, 568-581.

Dethloff, J. (2001) Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spektrum* 23, 79-96.

Gajpal, Y., Abad, P. (2009) An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research* 36, 3215-3223.

Hansen, P., Mladenović, N. (2001) Variable neighborhood search: principles and applications. European Journal of Operational Research 130, 449-467.

Jun, Y., Kim, B.-I. (2012) New best solutions to VRPSPD benchmark problems by a perturbation based algorithm. *Expert Systems with Applications* 39, 5641-5648.

Min, H. (1989) The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A* 23, 377-386.

Mladenović, N., Hansen, P. (1997) Variable neighborhood search. *Computers & Operations Research* 24, 1097-1100.

Nagy, G., Salhi, S. (2005) Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* 162.

Ropke, S., Pisinger, D. (2006) A unified heuristic for vehicle routing problems with backhauls. European Journal of Operational Research 171, 750-775.

Salhi, S., Nagy, G. (1999) A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society* 50, 1034-1042.

Subramanian, A., Cabral, L.d.A.F. (2008) An ILS Based Heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery and Time Limit. *Lecture Notes in Computer Science* 4972, 135-146.

Subramanian, A., Drummond, L.M.A., Bentes, C., Ochi, L.S., Farias, R. (2010) A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research* 37, 1899-1911.

Wassan, N.A., Wassan, A. H., Nagy, G. (2008) A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of Combinatorial Optimization* 15, 368-386.

AN EVOLUTIONARY LOCAL SEARCH WITH THRESHOLD ACCEPTING FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

Christian Prins

ICD–LOSI, Troyes University of Technology, BP 2060, 10010 Troyes, France, christian.prins@utt.fr

ABSTRACT

A new metaheuristic called MS-ELS-TA and derived from iterated local search (ILS) is proposed for the capacitated vehicle routing problem (CVRP). The method can be viewed as a kind of multi-start evolutionary local search (ELS), an ILS in which each iteration generates several children-solutions using mutation and local search. Moreover, the search alternates between solutions encoded as TSP tours (vehicle capacity is ignored) and CVRP solutions, and children may exceed the cost of their parent by a threshold value which is progressively reduced during the iterations. The resulting algorithm is evaluated on two sets of standard CVRP instances and competes with the seven best metaheuristics published.

Keywords: vehicle routing problem, evolutionary local search, threshold accepting

1 INTRODUCTION

The well-known *Capacitated Vehicle Routing Problem* (CVRP) is widespread in distribution logistics. This NP–hard problem is usually defined on a complete undirected network $G = (V, E)$ with a node set $V = \{0, 1, \ldots, n\}$ and an edge set E. Node 0 is a depot with identical vehicles of capacity Q. Each other node i represents a customer with a demand q_i and each edge (i, j) has a traversal cost $c_{ij} = c_{ji}$.

The CVRP consists in determining a set of vehicle trips of minimum total cost, such that each trip starts and ends at the depot, each customer is visited exactly once, and the total demand handled by a vehicle does not exceed Q. In some classical instances, each customer has a service cost s_i and the total cost of each trip may not exceed a fixed limit L. In the literature on metaheuristics, the number of vehicles used is a decision variable.

The reader will find in Golden *et al*. (2008) a recent state of the art on the CVRP. The most effective exact method (Baldacci and Mingozzi, 2008), can solve consistently instances with 100 customers. As many real instances are much larger, the development of heuristic algorithms remains essential.

More than 50 CVRP metaheuristics have been published since the 90's, inducing a strong international competition. The seven best considered in this paper are quite recent, these are: the evolution strategy of Mester and Bräysy (2007), the two memetic algorithms of Nagata and Braÿsy (2009) and Vidal *et al*. (2011), the tabu search–guided local search of Zachariadis and Kiranoudis (2010), the parallel record–to–record travel algorithm of Groër and Golden (2011), the bee mating method of Marinakis and Marinaki (2011), and the cooperative tabu search of Jin *et al*. (2010).

The aim of this talk is to design an improved variant of a multi–start ELS introduced by Prins in 2009 and to compare it with the top-seven methods cited above. Sections 2 and 3 respectively recall the principles of ILS and ELS and the components used in our MS–ELS of

2009. Section 4 describes the new version, which is compared to the competitors in Section 5, using two sets of classical instances with 50 to 483 customers.

2 ILS, ELS AND MS–ELS

To introduce our metaheuristic in a simpler way, we first recall the principles of Iterated Local Search (ILS), Evolutionary Local Search (ELS) and Multi–Start ELS (MS–ELS). In the sequel, $f(S)$ and BS respectively denote the cost of a solution S and the current best solution.

ILS (Lourenço *et al*., 2002) is a simple metaheuristic sketched in Algorithm 1. It requires three components: an initial greedy heuristic H, a mutation procedure *Mutate* and a local search (improvement procedure) *LS*. The initial solution is generated by H and improved using the local search. Then ILS performs a given number of iterations ni. Each iteration applies mutation and local search to a copy S of BS. BS is updated only if the resulting child-solution is better. Note that ILS generates a sequence of local optima with decreasing costs.

```
H (BS)
LS (BS)
for i := 1 to ni do
    S := BS
    Mutate (S)
    LS (S)
    if f (S) < f (BS) then BS := S
endfor
```

Algorithm 1. ILS template

ELS was introduced by Wolf and Merz (2007) to solve two optimization problems in telecommunications. The difference with ILS is to generate nc children at each iteration and to update BS only if the best child is better. Note that ILS is an ELS with $nc = 1$.

Prins (2009) designed for the CVRP a multi–start ELS (MS–ELS), in which H is replaced by a randomized heuristic *HR* and the *ELS* is nested in a main loop doing np iterations called phases. This external loop is used for diversification, launching successive ELS from initial solutions well spread in solution space.

In ILS, ELS and MS–ELS, typically 90% of CPU time is spent in the local search. The number of LS calls or "computing "budget" is roughly $B = np \times ni \times nc$. By playing on these parameters, we can retrieve three particular cases:

- if $np = nc = 1$, MS-ELS reduces to ILS.
- if $np = 1$, we get an ELS.
- if $ni = nc = 0$, we get a GRASP (Greedy Randomized Adaptive Search Procedure), in which np independent solutions computed by *HR* are improved by a single call to *LS*.

As shown by Prins (2009) for a given budget B, GRASP and ILS are slower and less effective than variants in which the budget is spread over the different loops. For instance, for $B = 25000$, a version with $np = 10$ initial solutions, $ni = 50$ iterations per ELS and $nc = 50$ children per ELS iteration is better than a pure ILS or a pure GRASP with 25000 iterations.

3 EXISTING MS–ELS FOR THE CVRP

3.1 Alternation between two search spaces

Apart from a MS–ELS structure introduced in Section 2, a key–feature of the CVRP metaheuristic in (Prins, 2009) is to alternate between two search spaces: the space of giant tours and the space of CVRP solutions. A giant tour is in fact a TSP tour covering the depot and all customers (vehicle capacity is ignored). Any giant tour T can be converted in $O(n^2)$ into an optimal CVRP solution S (subject to the sequence), using a splitting procedure *Split*.

Conversely, we can reconstitute a giant tour with a procedure *Concat* that concatenates the sequences of customers of each trip in a given CVRP solution. It is easy to show that there exists an optimal giant tour, i.e., one giving an optimal CVRP solution after splitting. This is determinant for the efficiency of the method: without any loss of information, the MS–ELS can browse the set of giant tours, which is much smaller than the set of CVRP solutions.

Due to the limited number of pages, we do not detail here the 2009 algorithm. However its structure is included in the algorithm given in Section 4 for the new version. We just remind how the alternation is added to the general MS–ELS model of Section 2.

Instead of working on an incumbent CVRP solution S, MS-ELS considers a pair (S, T), where T is the giant tour obtained by concatenating the trips of S. The randomized heuristic generates a giant tour instead of a *CVRP* solution. This tour is split to get a feasible CVRP solution S, which is improved by local search. Finally, the trips are concatenated to return the giant tour associated with S, giving the initial pair (S, T) for each ELS. At each ELS iteration, *nc* children are generated. To obtain a child, a copy T' of the giant tour T is mutated and then split to get a CVRP solution S' which undergoes the local search. The current solution S is updated if the best child improves it. The trips of S are finally concatenated to yield the pair (S, T) for the next ELS iteration.

3.2 Components of MS–ELS

We recall the components of the existing MS–ELS as they are reused in the new variant.

Randomized heuristic HR(T). It is extremely simple and corresponds to a randomized version of the nearest–neighbour heuristic for the TSP. For an emerging trip ending at customer i, *HR* determines the K nearest customers not yet visited and selects randomly one of them to add it after i. In all our experiments, $K = 2$ is enough to get diversified but fairly good solutions after splitting.

Mutation procedure Mutate(T). It consists in performing m random swaps of two customers in the giant tour. The “mutation level” m is dynamically adjusted between 1 and a maximum value *maxm*. Starting from 1, m is incremented if the current ELS iteration fails to improve the parent solution (no child is better), otherwise it is decremented. Note that these perturbations could violate vehicle capacity or be easily repaired by the local search if they were applied to CVRP solutions. On a giant tour, there is no feasibility problem and even one single swap can lead to a different solution after splitting.

Splitting procedure Split(T,S). Roughly speaking, an auxiliary acyclic graph H with n+1 nodes 0, 1, …, n is built. Each subsequence (T_i, T_{i+1}, …, T_j) of customers in the giant tour T which can be visited in this order by a feasible trip is modelled in H by one arc (i-1, j), weighted by the trip cost. The optimal splitting, subject to the sequence imposed by T, is obtained by computing a shortest path from node 0 to node n in H. An example with a figure and a detailed algorithm that does not build H explicitly can be found in Prins (2004).

Local search procedure LS(S). Applied to CVRP solutions, it is based on three classical moves: 2–opt moves that replace two edges by two others, Or–opt moves that relocate a string of customers, and string exchanges that swap two strings of customers. The strings may have one to three customers and can have different lengths in the string exchanges. Moreover, all moves may involve one or two trips, e.g., a string of two consecutive customers can be removed and reinserted at a different position, in the same trip or in another route. We use a fast implementation proposed by Irnich *et al*. (2006), in which each move is decomposed into a sequence of partial moves. Many sequences are pruned using lower and upper bounds.

Concatenation procedure Concat(S,T). This step consists simply in concatenating the sequences of customers of each trip of S to get a giant tour T.

4 THE NEW VARIANT MS–ELS–TA

There are two main differences between the 2009 algorithm and the new variant called MS–ELS–TA (MS–ELS with threshold accepting).

The first difference is to add a deterministic annealing scheme called *threshold accepting* (Dueck and Scheuer, 1990). A degradation factor α is defined. At the beginning of each ELS, α is set to a maximum value α_0. After each ELS iteration, the parent S is replaced by the best child S' if $f(S')$ is smaller than the threshold $(1+\alpha).f(S)$. For instance, $\alpha = 0.1$ means that degradations of up to 10% are accepted. The factor α decreases linearly with the iterations, to cancel after $\beta.ni$ iterations, $0 < \beta < 1$. If $\beta = 2/3$ for instance, the algorithm rejects non-improving children after two thirds of iterations and then behaves like the 2009 algorithm.

```
nls := 0                                // number of calls to local search
while nls < np.ni.nc do                 // main loop (multi-starts) with a budget np.ni.nc
    HR (T)                              // compute one random initial giant tour T
    Split (T,S)                         // split it to get a CVRP solution S
    LS (S)                              // apply local search to CVRP solution
    nls := nls + 1                      // and count one call
    if nls = 1 then BS := S             // record initial global best soln BS
    nat := 0                            // initial nb of ELS iterations above threshold
    m := 1                              // initialize mutation level
    α := α0                             // initial degradation factor used for threshold
    for i := 1 to ni do                 // ELS loop
        Concat (S,T)                    // build giant tour corresponding to S
        f' := (1+α).f(S)                // initialize cost of best child S' to threshold
        for j := 1 to nc do             // loop generating the nc children
            T" := T                     // take a copy of current giant tour
            Mutate (T", m)              // apply mutation with mutation level m
            Split (T", S")              // split mutated tour to get a CVRP soln S"
            LS (S")                     // apply local search to CVRP soln
            nls := nls + 1              // count one call
            if f(S") ≤ f' then          // if child S" outperforms best child
                S' := S"                // then update best child
                f' := f(S')             // and its cost
            endif
        endfor                          // end of loop for children
        α := α - α0 / (β.ni)            // decrease degradation factor
        if f' < (1+α).f(S) then         // if best child below threshold
            S := S'                     // replace parent by best child
            m := max(m-1,1)             // decrement mutation level (no less than 1)
            nat := 0                    // reset nb of iterations above threshold
        else                            // otherwise (no child below threshold)
            m := min(m+1,maxm)          // increment mutation level (no more than maxm)
            nat := nat + 1              // count one iteration above threshold
        endif
        if f' < f(BS) then BS := S'     // update global best solution
        if nat = maxnat then break      // drop ELS after maxnat iterations above threshold
    endfor
endwhile
```

Algorithm 2. General structure of MS-ELS-TA.

Concerning the second difference, we observed that the ELS is often blocked well before the *ni* allocated iterations. So, we prefer to stop the ELS earlier, after a given number *maxnat* of consecutive iterations without getting a child below threshold. However, we keep the same budget $B = np.ni.nc$ for the number of calls to the local search, which allows MS–ELS–TA to perform more than *np* restarts, as long as its budget is not exhausted.

In other words, from iterations 1 to $\beta.ni$, cost degradations are accepted if they stay below the threshold, the trajectory may go up and down, and the ELS stops after *maxnat* consecutive iterations failing to find a child below threshold. From iterations $\beta.ni + 1$ to *ni*, degradations are forbidden and the trajectory must go down (like in the 2009 version), but the ELS may stop earlier after *maxnat* iterations blocked on the same parent. The idea is that it is more fruitful to restart the ELS in an unexplored region rather than losing time in unproductive iterations.

The resulting metaheuristic is summarized and commented in Algorithm 2.

5 COMPUTATIONAL EVALUATION

5.1 Implementation, instances and competitors considered

The proposed metaheuristic was implemented in Delphi and tested on a Dell portable computer with an Intel Core i7 processor at 2.20 GHz and Windows 7 Professional.

Two classical sets of Euclidean benchmark problems were used. The first one (Christofides *et al*., 1979) contain 14 instances with 50 to 199 customers. Instances 6 to 10, 13 and 14 have non-zero service times and a maximum trip duration. The second set (Golden *et al*., 1998) provides 20 larger instances with 200 to 483 customers. Instances 1 to 8 have a maximum trip duration but service times are null. To get the two sets, just google “VRP web”.

The competitors considered are the seven best metaheuristics for the second set, already cited in the introduction: the evolution strategy (ES) of Mester and Bräysy (2007), the two memetic algorithms (MA) of Nagata and Braÿsy (2009) and Vidal *et al*. (2011), the tabu search–guided local search (TS-GLS) of Zachariadis and Kiranoudis (2010), the parallel record–to–record travel algorithm (RTR) of Groër and Golden (2011), the bee mating method of Marinakis and Marinaki (2011), and the cooperative tabu search and Jin *et al*. (2010).

5.2 Results for the Christofides *et al*. instances

We used the following parameters after a design of experiments that cannot be detailed here due to limited space: number of phases (ELS) $np = 12$, iterations per ELS $ni = 80$, children per ELS iteration $nc = 100$, maximum degradation factor $\alpha_0 = 0.1$, degradation factor cancelled after a fraction $\beta = 2/3$ of the *ni* iterations, number of consecutive ELS iterations allowed above threshold $maxnat = 5$, maximum number of swaps in the dynamic mutation $maxm = 3$. Only one run was executed for each instance: the average value on more runs is very close, because our method is already based on restarts.

Table 1 lists the solution values of MS-ELS (Prins, 2009) and the new variant with threshold accepting MS-ELS-TA. The first column indicates the instance number and the number of customers *n*. The second column indicates whether the instance has a maximum trip duration (MTD) or not. The third column provides the best known solutions (BKS) gathered in published papers. Note that no published algorithm is able to find all these BKS simultaneously. The four next columns give for our two algorithms the solution cost and the time in seconds to reach these solutions on our PC. Costs equal to the BKS are in boldface. The last column shows the best solutions found by MS-ELS-TA during our tests, using various sets of parameters. The last line displays average values for the gaps to BKS (in percent) and the running time.

The table shows that MS-ELS-TA divides by 2.7 the average gap of MS-ELS and finds 12 BKS instead of 10, at the expense of a doubled running time. However, allocating more time to MS-ELS does not improve its results significantly.

Table 1. Detailed results of MS-ELS and MS-ELS-TA on Christofides *et al*. instances

File-*n*	MTD	BKS	MS-ELS	Time (s) [1]	MS-ELS-TA	Time (s) [2]	Our best
01-050	N	524.61	**524.61**	0.06	**524.61**	0.02	**524.61**
02-075	N	835.26	**835.26**	15.01	**835.26**	10.07	**835.26**
03-100	N	826.14	**826.14**	3.19	**826.14**	1.30	**826.14**
04-150	N	1028.42	1029.48	44.84	**1028.42**	71.97	**1028.42**
05-199	N	1291.29	1294.09	11.85	1291.50	5.30	*1291.50*
06-050	Y	555.43	**555.43**	0.14	**555.43**	0.11	**555.43**
07-075	Y	909.68	**909.68**	5.18	**909.68**	0.94	**909.68**
08-100	Y	865.94	**865.94**	0.32	**865.94**	0.13	**865.94**
09-150	Y	1162.55	**1162.55**	6.58	**1162.55**	2.11	**1162.55**
10-199	Y	1395.85	1401.46	125.57	1400.67	230.30	*1399.85*
11-120	N	1042.11	**1042.11**	0.64	**1042.12**	0.25	**1042.11**
12-100	N	819.56	**819.56**	0.17	**819.56**	0.09	**819.56**
13-120	Y	1541.14	1545.43	9.25	**1541.14**	142.14	**1541.14**
14-100	Y	866.37	**866.37**	0.27	**866.37**	0.11	**866.37**
Average			0.071%	15.93s	0.026%	33.20s	0.022%

1: Intel Pentium 4 2.8 GHz, 2: Intel Core i7 2.20 GHz

In Table 2, our two algorithms are compared with the selected competitors, except Zachariadis and Kiranoudis and Jin *et al*., who have not tackled this set of instances. The table recalls the authors, the year of publication and the kind of method, before giving the average gap to BKS in %, the average duration in minutes, the processor used and the testing conditions. We wished to scale the running times but gave up because the powers which can be found in the literature are not precise: they rely on specific calculations (matrix operations) and are strongly affected by the language and compiler used, the compilation options, the memory and cache technology etc. According to a private discussion with Eric Taillard, the running time can be halved or doubled if a program is rewritten in another language on the same machine, or moved to a computer with the same clock speed but a different processor.

Table 2. Comparison of MS-ELS-TA with competitors on Christofides *et al*. instances

Authors	Year	Method	Gap	Time (min)	Processor	Runs
Marinakis-Marinaki	2011	Bee mating	0.009%	50×0.74	Pentium M 1.86 GHz	Best 50
Vidal et al.	2011	Slow MA	0.020%	9.76	AMD Opteron 2.4 GHz	Avg 10
Prins	2012	MS-ELS-TA	0.026%	0.55	Intel Core i7 2.2 GHz	Single
Mester-Bräysy	2007	ES	0.027%	2.71	Intel Pentium 4 2.8 GHz	Single
Gröer-Golden	2011	Parallel RTR	0.028%	8×25	8 × Intel Xeon 2.3 GHz	Best 5
Nagata-Bräysy	2009	HGA	0.033%	1.60	AMD Opteron 2.4 GHz	Avg 10
Vidal et al.	2010	Std MA	0.047%	2.21	AMD Opteron 2.4 GHz	Avg 10
Prins	2009	MS-ELS	0.071%	0.27	Intel Pentium 4 2.8 GHz	Single

Gröer and Golden test their parallel RTR on 4, 8 and 129 processors, using 5 runs of 5 minutes each (wall clock time). They get the best gap for 129 processors (0.001%) in 25 minutes but the total time is huge: 5×5×129 = 3225 minutes! For a fair comparison with other metaheuristics, these authors recommend their results with 8 processors: in that case the gap becomes 0.027% (rank 5 in the table) and the total time is still large (200 minutes). The bee mating approach of Marinakis and Marinaki provides the best gap but requires a large running time too. Vidal et al. (2011) evaluate a standard version (rank 7) and a slower version with five times more iterations (rank 2). The MS-ELS-TA can be found at rank 3, with a running time one order of magnitude faster than the two better algorithms, followed by the ES, the RTR and the two MA. Our 2009 MS-ELS is at the end but displays the smallest running time.

To conclude with these instances, MS-ELS-TA is faster than the other methods. The two metaheuristics able to produce better results last at least 18 times longer. However, it is fair to say that all gaps are now very small, even if no method is able to find all BKS. Moreover, apart from the RTR, the slow MA and the bee mating method that are one order of magnitude slower, the difference in duration between the other algorithms is not very significant.

5.3 Results for the Golden *et al*. instances

These results are listed in Tables 3 and 4, using the same formats as Tables 1 and 2. In our opinion, these large and highly symmetric instances with concentric rings of customers are unrealistic but, unfortunately, they are used by all recent metaheuristics. They affect all local search procedures, which perform a lot of moves, with gains as small as 10^{-3}. Hence, the following parameters were changed to follow the slower convergence of each ELS: iterations per ELS $ni = 250$, maximum number of consecutive iterations above threshold $maxnat = 30$, maximum mutation level $maxm = 2$. There are so many improving (but small) moves that it was not required to mutate as strongly ($maxm = 3$) as for the previous set of instances. The other parameters were kept: number of phases (ELS) $np = 12$, number of children per ELS iteration $nc = 100$, maximum degradation factor $\alpha_0 = 0.1$, degradation factor cancelled after a fraction $\beta = 2/3$ of the ni iterations. Here again, one run was executed for each instance.

Table 3. Detailed results of MS-ELS and MS-ELS-TA on Golden *et al*. instances

File-*n*	MTD	BKS	MS-ELS	Time (s) [1]	MS-ELS-TA	Time (s) [2]	Our best
1-240	Y	5623.47	5644.52	238.02	5624,93	306.86	**5623.47**
2-320	Y	8404.61	8447.92	93.12	8423.27	1432.51	8423.27
3-400	Y	11036.22	**11036.22**	384.94	**11036.22**	434.97	**11036.22**
4-480	Y	13592.88	13624.52	349.82	13624.52	1070.53	13624.52
5-200	Y	6460.98	**6460.98**	62.76	**6460.98**	0.47	**6460.98**
6-280	Y	8404.26	8412.90	84.44	8412.90	51.99	8412.90
7-360	Y	10102.70	10195.59	463.21	10195.59	246.81	10195.59
8-440	Y	11635.30	11643.90	298.05	11643.90	1573.93	11635.34
9-255	N	579.71	586.23	78.62	583.49	1037.18	583.19
10-323	N	736.62	744.36	437.87	743.82	610.22	740.29
11-399	N	913.22	922.40	272.82	920.95	2023.45	917.96
12-483	N	1102.76	1116.12	1746.33	1110.85	736.92	1110.85
13-252	N	857.19	862.32	128.69	863.24	134.32	859.47
14-320	N	1080.55	1089.35	546.93	1081.31	885.74	1080.17
15-396	N	1337.92	1352.39	613.36	1349.43	411.20	1343.59
16-480	N	1612.50	1634.27	1262.99	1628.67	1188.31	1625.66
17-240	N	707.76	708.85	61.63	707.79	737.65	**707.76**
18-300	N	995.13	1002.15	175.55	998.52	1429.85	998,52
19-360	N	1365.60	1371.67	1078.54	1370.58	426.27	1367.40
20-420	N	1818.32	1830.98	346.49	1826.36	566.98	1825.70
Average			0.623%	436.21	0.429%	765.31	0.320%

1: Intel Pentium 4 2.8 GHz. 2: Intel Core i7 2.20 GHz

Table 3 shows that MS-ELS-TA reduces the gap by one third compared with MS-ELS, but the running time grows by 80%. MS-ELS-TA is at least as good as MS-ELS, except for instance 13. Here again, we tried to allocate more iterations to MS-ELS, to have the same running time for the two methods, but the gap reduction was negligible.

The comparison with the other metaheuristics in Table 4 indicates that the best results are obtained by memetic algorithms (Vidal et al., 2011; Nagata and Braÿsy, 2009). MS-ELS-TA can be found at rank 7. All running times increase strongly as compared with the previous set of instances. In terms of total CPU time, our two algorithms are the fastest ones. They are even one order of magnitude faster than the two parallel implementations (Gröer and Golden, 2011; Jin et al., 2010) and the bee algorithm (Marinakis and Marinaki, 2011).

Even if the average gaps of all compared heuristics are rather small, below 1%, these instances are probably not yet closed, as shown by the improvement of most BKS in the last three years.

We think that we can improve our results further in the near future and that MS-ELS-TA offers a good compromise between solution quality, running time and simplicity. Indeed, the proposed algorithm is based on simple components (randomized heuristic, mutation, local search and deterministic annealing). It does not require a sophisticated crossover like in Nagata and Braÿsy (2009), penalized infeasible solutions or elaborated diversification schemes like in Vidal et al. (2011), or complicated data structures like the Fibonacci heaps used by Zachariadis and Kiranoudis (2011). Only the RTR of Gröer and Golden (2011) looks simpler, except its parallel implementation. Moreover, MS-ELS-TA requires relatively few parameters (7), to be compared for instance to 12 parameters in the evolutionary strategy of Mester and Braÿsy (2007).

Table 4. Comparison of MS-ELS-TA with competitors on Golden *et al*. instances

Authors	Year	Method	Gap	Time (min)	Processor	Runs
Vidal et al.	2011	Slow MA	0.15%	71.41	AMD Opteron 2.4 GHz	Avg 10
Vidal et al.	2011	Std MA	0.26%	21.57	AMD Opteron 2.4 GHz	Avg 10
Nagata-Bräysy	2009	HGA	0.27%	21.51	AMD Opteron 2.4 GHz	Avg 10
Gröer-Golden	2011	Parallel RTR	0.29%	8×25	8 × Intel Xeon 2.3 GHz	Best 5
Mester-Bräysy	2007	ES	0.32%	24.40	Intel Pentium 4 2.8 GHz	Single
Zachariadis-Kiranoudis	2010	GLS-TS	0.42%	40.50	T5500 1.6 GHz	Avg 10
Prins	2012	MS-ELS-TA	0.43%	12.76	Intel Core i7 2.2 GHz	Single
Marinakis & Marinaki	2011	Bee mating	0.55%	50×3.96	Pentium M 1.86 GHz	Best 50
Jin et al.	2010	Coop. tabu	0.59%	8×41.9	2 Xeon 3.0 GHz, 8 cores	Avg 10
Prins	2009	MS-ELS	0.62%	7.27	Intel Pentium 4 2.8 GHz	Single

6 CONCLUSION

This talk presents a new metaheuristic MS-ELS-TA, inspired by ILS and able to compete with the top-seven metaheuristics for the CVRP. Even if MS-ELS-TA does not become the most efficient approach, it displays a good compromise between solution quality, running time and simplicity. Our approach does not involve a population of solutions, sophisticated crossovers operators, ruin and recreate moves, or elaborated diversification techniques. However, it is based on an original feature which is probably the key for its efficiency: the alternation between two search spaces (giant tours and CVRP solutions).

The new algorithm contributes to the race in progress for better CVRP methods, a race that will continue as long as the problem stays out of reach of exact methods for more than 100 nodes. We are now testing the same general structure on other vehicle routing problems.

REFERENCES

Baldacci, R., Christofides, N., and Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts, *Mathematical Programming*, **115**, 351–385.

Christofides, N., Mingozzi, A., and Toth, P. (1979). The vehicle routing problem. In: N. Christofides, A. Mingozzi, P. Toth and C. Sandi (eds), *Combinatorial optimization*, Wiley, Chichester, pp. 315–338.

Dueck, G., and Scheuer, T. (1990). Treshold accepting: A general purpose optimization algorithm, *Journal of Computational Physics*, **90**, 161–175.

Golden, B.L., Wasil, E.A., Kelly, J.P., and Chao, I.M. (1998). The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets and computational results. In: T.G. Crainic and G. Laporte (eds), *Fleet management and logistics*, Kluwer, Dordrecht, pp. 33–56.

Golden, B.L., Raghavan, S., and Wasil. E.A. (2008). The vehicle routing problem: latest advances and new challenges, Springer.

Groër, C., and Golden, B.L. (2011). A parallel algorithm for the vehicle routing problem, *INFORMS Journal on Computing*, **23**, 315–330.

Irnich, S., Funke, B., and Grünert, T. (2006). Sequential search and its application to vehicle routing problems, *Computers & Operations Research*, 33, 2405–2429.

Jin, J., Crainic, T.G., and Lokketangen, A. (2010). A parallel multi-neighborhood cooperative tabu search for the capacitated vehicle routing problem, Research Report 2010-54, CIRRELT, Montréal.

Lourenço. H.R., Martin. O., and Stützle, T. (2003). Iterated local search. In: F. Glover and G. Kochenberger (eds), *Handbook of metaheuristics*, Kluwer, Dordrecht, pp. 321–353.

Marinakis, Y., and Marinaki, M. (2011). Bumble bees mating optimization algorithm for the vehicle routing problem. In: B.K. Panigrahi, Y. Shi, M.H. Lim, L.M. Hiot, Y.S. Ong (eds), *Handbook of Swarm Intelligence, Adaptation, Learning and Optimization*, vol. 8, Springer, Berlin, pp. 347–369.

Mester, D., and Bräysy, O. (2007). Active-guided evolution strategies for large-scale capacitated vehicle routing problems, *Computers & Operations Research*, **34**, 2964–2975.

Nagata, Y., and Bräysy, O. (2009). Edge-assembly_based memetic algorithms for the capacitated vehicle routing problem, *Networks*, **54**, 205–215.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers & Operations Research*, 31, 1985–2002.

Prins, C. (2009). A GRASP x Evolutionary local search hybrid for the vehicle routing problem. In: F.B. Pereira and J. Tavares (eds), *Bio-inspired algorithms for the vehicle routing problem*, Springer, Berlin, pp. 35–53.

Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., and Rei, W. (2011). A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems, Research Report 2011-05, CIRRELT, Montréal. To appear in *Operations Research*.

Wolf, S., and Merz, P. (2007). Evolutionary local search for the super-peer selection problem and the p-hub median problem. In: T. Bartz-Beielstein, M.J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph and M. Sampels (eds), *Hybrid metaheuristics 2007*, LNCS 4771, Springer, Berlin, pp. 1–15.

Zachariadis, E.E., and Kiranoudis. C.T. (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem, *Computers & Operations Research*, **37**, 2089–2105.

A FIXED ZONES PARCEL DISTRIBUTION STRATEGY IN THE CASE OF EXCESS DEMAND

Nenad Bjelić[1], Dražen Popović[1], Milorad Vidović[1] and Gordana Radivojevic[2]

[1]University of Belgrade, Vojvode Stepe 305, 11000 Belgrade, Serbia, n.bjelic@sf.bg.ac.rs, d.popovic@sf.bg.ac.rs, mvidovic@sf.bg.ac.rs

[2]University of Belgrade, Institute Mihajlo Pupin, Volgina 15, Belgrade 11000, Serbia, gordana.radivojevic@pupin.rs

ABSTRACT

Due to the decision of large number of companies to outsource distribution part of business, parcel distribution companies (PDC) are faced with large amount of pick-ups and deliveries (P&D) on every day basis. However, the biggest challenge PDCs are facing with is stochastic nature of distribution process in terms of P&D locations, sizes and weights. Besides that, another important aspect of PDCs' operation is consistent service of clients with the same drivers. The most important reason for this consistency requirement is service efficiency due to layout familiarity of a driver in consistently served zones, as well as due to the trust relations established between driver and customer during the time which eventually leads to reduction of P&D service times. In this paper we compare efficiencies of two operational planning strategies for realizing delivery part of distribution process in situation when overall capacity of distribution vehicles is exceeded and additional vehicle must be introduced. The first strategy is well known strategy of using sweep vehicle to service excess of demand in each delivery zone. The second strategy is new one and it implies minimization of both inconsistent deliveries, i.e. deliveries made by not consistent driver (vehicle) and tour lengths.

Keywords: parcel distribution, operational planning, VND

1 INTRODUCTION

The need to focus on the company's core business in order to gain competitive advantages in the quality of the finished product, the increase of product distribution costs, caused by the increase in oil prices, and the aspiration to expand the market to larger geographical regions, has led to a situation in which an increasing number of companies decide to outsource the distribution of products to specialized distribution companies. Accordingly, Parcel Delivery Companies (PDC) have gradually changed their original business concepts from mostly customer-to-customer (C2C) market segment to business-to-business (B2B) and business-to-customer (B2C) services. For this reason it is not surprisingly that the distribution configuration has fewer pick-up than delivery locations. Therefore, because more effort should be made to realize delivery tasks, it can be concluded that the potential for improving the distribution system is much higher on the side of the parcel delivery, then on the side of parcel collection.

The basic service of PDC is delivery within 24 hours and it is accomplished by implementing hub and spoke transport organization with two main segments: long haul and local transport. Long haul transport uses high-capacity trailers for transferring shipments between hub and local distribution centers. Local transport is performed by vehicles of a smaller capacity used for parcel deliveries and pick-ups on the territory of a local distribution center. The The territory of each local distribution center, or depot, as it is referred in the following of the paper, is divided into zones. At the beginning of a working day all parcels collected from a previous day (i.e. parcels received from the hub) must be delivered by vehicles assigned to specified zones. During the process of parcel delivery, PDC receives pick-up demands that are being batched and serviced after completion of the last delivery task. Parcels being picked-up are delivered on the next day. In the case when parcel pick-up and delivery locations are in the same depot area, parcel is not sent to the hub for sorting. Otherwise, parcels are sent to the hub where they are sorted and during the night transferred to a delivery depot, where they are prepared for delivery during the next working day.

Although the routing of vehicles in the area divided in zones is not adaptable to the stochastic nature of parcels' pick-up and delivery requirements (i.e. delivery locations, volumes and weights may vary considerably from day to day), courier companies insist on this kind of organization (Sorensen *et al.* 2008.). Main reason for that lies in time savings related with implementation of zone based organization. Namely, driver's knowledge of a zone layout and his familiarity with the traffic conditions in a certain area decreases a time driver needs to spend reaching next delivery location. In addition, the procedures to be complied with B2B clients are significantly reduced by mutual driver to client trust when a client is serviced consistently by the same driver.

Based on previously said, due to higher intensity of delivery demands compared to pick-up ones, in this paper we focus on improvement of delivery part of distribution process in a PDC. We suppose that depot area is separated in several fixed zones with a delivery vehicle assigned to serve each zone. However, because of stochastic demand, not so rarely PDC comes into situation where delivery demand exceeds capacity of the vehicle assigned to cretin zone, or even exceeds overall capacity of available vehicles. Therefore, in order to provide expected quality of service PDC needs to develop some strategy to avoid or decrease negative consequences of this kind of disturbances. In this paper we propose combined strategy based on delivery tasks redistribution and sweep vehicle introduction, under objective of driver to customer service consistency and total routes length minimization.

Paper is organized in such way that in the following section we give detailed problem description, present two strategies for operational planning in conditions of excess demand and review the relevant literature. In section 3 we provide mathematical model for new operational planning strategy by using MILP formulation. In section 4 we introduce easy to implement four step heuristic algorithms, while in section 5 we preset results obtained by implementing proposed algorithm on set of generated problem instances. Finally, at the end of paper we give conclusion.

2 PROBLEM DESCRIPTION AND LITERATURE OVERVIEW

In PDCs implementing fixed zone delivery strategy, stochastic nature of the process regarding deliveries' locations, volumes and weights is the main reason why it is very difficult to determine the borders of each zone so that assigned vehicles can satisfy all demands each working day. Accordingly, it is to be expected that some vehicles will be over and some other under capacitated. Therefore, in cases when overall vehicles' capacity exceeds total demand, but there are certain zones with insufficient vehicle capacity, some level of task sharing between zones could be an option to overcome that imbalance. Another possible consequence of stochastic demand is that even with sharing between zones, overall vehicles' capacity can not satisfy total delivery requirement. In other words, demand is greater than the total capacity of all assigned vehicles. Solution for this problem can be introduction of additional vehicle(s) not assigned to any fixed zone. Additional (sweep) vehicle is used for service of all residual demands i.e. demands that can not be satisfied with vehicles assigned to fixed zones. A typical way of engaging the sweep vehicle is presented in Fig. 1a. In the rest of the paper this strategy is referred as "typical sweep strategy" (TSS).

Implementation of the TSS may results in long sweep vehicle route, as well as in the reduction of delivery efficiency, because the driver isn't familiar with the entire territory served by the depot. Therefore, with the objective of reducing the total length of delivery routes, and especially the length of a sweep route, in this paper we present a new strategy for redistribution of tasks (Consistent Delivery Minimal Reassignment Strategy - CDMRS) between already existing routes. CDMRS allows vehicles with residual capacity to cross into zones with demands exceeding vehicle capacities in order to serve unsatisfied demands. By allowing vehicles to serve tasks in non-assigned zones CDMRS tries to reduce the number of zones in which sweep vehicle operates, i.e. to group those zones near the depot (see Fig. 1b). The reason for grouping all excess demands near depot is the reduction of the sweep route length and intention to preserve driver to customer assignment consistency. However, because driver to customer consistency in other routes in this way is also disrupted, an important aspect of CDMRS is its minimization.

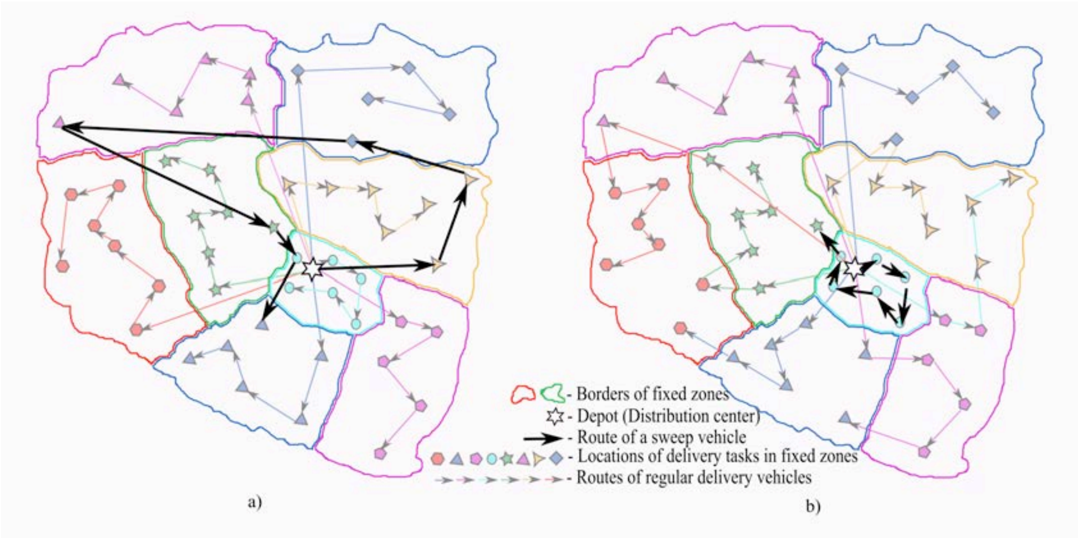


Figure 1. Task service in inconsistent zones: (a) allowed only to sweep delivery vehicle, and (b) allowed to all vehicles

Precisely, application of CDMRS increases the length of regular routes. Nevertheless, that is compensated in sweep route length reduction, meaning that CDMRS incurs decrease in total length of all routes. On the other hand, drivers to customers consistency reduction means that decrease in overall length of routes does not imply increase in service efficiency because drivers need more time to serve tasks out of their zones. Therefore, minimization of the number of deliveries made by drivers which are outside of their zones is introduced as additional objective measure in CDMRS. Moreover, by introducing coefficients, briefly explained in section 3, we try to distinguish driver's diverse familiarity with different zones.

Operation of PDCs until recently didn't attract wider research attention despite the fact that this industry records constant growth in the last decade. In the available literature there are only two research papers considering parcel delivery problem with stochastic demands under the objective of maximizing driver to customer service consistency. Trade-off between use of fixed zones, providing consistent driver to customer service, and route flexibility, providing adjustment to stochastic nature of parcel distribution process, Zhong *et al.* (2007) realize by introducing two-stage vehicle routing model and the concept of "cell", "core area" and "flex zone". Based on historical data of frequencies and spatial distribution of task occurrences set of expected clients in planning horizon is grouped in the "cells", thus reducing the problem to much smaller size. In a strategic step, "cells" that have to be served with the same routes are grouped in the "core area". In operational step, based on information about demand during a specific day, the remaining "cells" are assigned to routes that are serving "core areas". At this step, in the case of the need for additional capacity, sweep vehicles are introduced to satisfy the excess demands. The concept of "flex zone" is used for balancing route loads. The idea of this concept is that the "cells" in close proximity to the depot are not assigned to any route in strategic step, but they are to be used in the operational steps in order to balance the load of all routes. Another distinguishing feature of this paper is use of learning and forgetting curves for evaluating effects of drivers' familiarity with visiting cells.

Schneider *et al.* (2012) developed a model for quantifying effects of fixed area based routing approach to the parcel delivery under time window constraints. The authors also propose a two-phase approach to obtain solution in the case of stochastic demands. In the districting phase, based on historical data, authors determine routes’ load balancing zone, “seed” customers and fixed customers which are located in a “seed” customer’s proximity. In the second phase, routing of vehicles every day is performed in such a way that all fixed customers of a “seed” customer must be served by the same vehicle, while the rest of customers is distributed on different routes with objective of maximizing driver to customer consistency. Because the introduction of new vehicles is not allowed in situations where time or capacitive constraints are not satisfied, proposed approach allows reassignment of the fixed customers between routs in order to obtain feasible solutions. In the case of uniformly distributed delivery locations, high variations in the number of delivery tasks and high density of time windows, proposed strategy showed moderate routes' distance increase, but gives good routes consistency.

Based on the importance that consideration of parcel delivery service has for PDCs, Wong R.T. (2008) proposes alternative approaches to the service territory design as one possible area of further research in parcel distribution related problems.

3 PROBLEM FORMULATION

In this paper we suppose that service zones are fixed and present the CDMRS approach which considers the problem on the operational level.

Presumptions for implementing CDMRS are:

1. Only delivery part of distribution process is considered
2. All data regarding delivery tasks are known prior to the beginning of implementing CDMRS
3. Overall quantity (number, volume or weight) of delivery tasks exceeds total capacity of all available vehicles, i.e. in order to preserve quality of service sweep vehicle must be engaged
4. Capacity of sweep vehicle is sufficient to satisfy all excess of demand

Let Z be a set of fixed zones $z \in Z$, with cardinality $|Z|=m$, and V be a set of all tasks $i \in V$, with the cardinality $|V|=n$, appeared during the observed interval which is usually one day, and distributed within zones. Also, let R be a set of vehicles $r \in R$, of cardinality $|R|=k+1$ which serve all delivery tasks. First k vehicles from the set R serve regular tasks' demand, while vehicle $k+1$represents sweep vehicle serving excess demand when exists. Note that because each vehicle makes single route, notation used for vehicles in the same time denote its routes. Note also, that in the regular system state, when no excess demand exist, number of vehicles is equal to the number of zones, i.e. $m=k$.

Then, CDMRS mathematical model can be formulated in the following way:

$$\min \sum_{i \in V} \sum_{z \in Z} \sum_{r \in R} b_{iz} u_{zr} y_{ir} + \xi \sum_{i \in V \cup \{0\}} \sum_{j \in V} \sum_{r \in R} c_{ij} x_{ijr} \tag{1}$$

Subject to

$$\sum_{r \in R} y_{ir} = 1 \quad \forall i \in V \tag{2}$$

$$\sum_{r \in R} y_{0r} = |R| \tag{3}$$

$$\sum_{i \in V \cup \{0\}} x_{ijr} = y_{jr} \quad \forall j \in V, \forall r \in R \tag{4}$$

$$\sum_{j \in V \cup \{0\}} x_{ijr} = y_{ir} \quad \forall i \in V, \forall r \in R \tag{5}$$

$$\sum_{i \in V} d_i y_{ir} \leq C_r \quad \forall r \in R \tag{6}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijr} \leq |S| - 1 \quad \forall S \subseteq V, |S| \geq 2, r \in R \tag{7}$$

$$y_{ir} \in \{0,1\} \quad \forall i \in V, \forall r \in R \tag{8}$$

$$x_{ijr} \in \{0,1\} \quad \forall i, j \in V \cup \{0\}, \forall r \in R \tag{9}$$

Where following notation is used:

b_{iz} - binary coefficient that denotes location of delivery task *i* within *z*, if $b_{iz} = 1$ task *i* is located within boundaries of zone *z*, otherwise it is located within some other zone.
y_{ir} - binary variable defining whether task *i* is served by vehicle *r*
u_{zr} - coefficient used to measure driver to customer service inconsistency. It represents decrease of service efficiency due to driver's *r* unfamiliarity with the zone *z*
ξ - sufficiently small value
c_{ij} - time or spatial distance between tasks *i* and *j*
x_{ijr} - binary variable defining whether vehicle *r* serves task *j* after task *i*
d_i - demand of task *i*
C_r - capacity of vehicle *r*

Increase in service inconsistency provides decrease of overall routes' length, i.e. the problem is eventually reduced to classical vehicle routing problem (VRP). Therefore, consideration of service consistency is the primal objective of CDMRS. For that reason, objective function (1) beside typical three index vehicle flow formulation (Toth and Vigo, 2002) contains part that respects driver to customer consistency. By using coefficient ξ emphasis in objective function is focused on minimization of inconsistency.

Product $b_{iz}y_{ir}$ equals one only if task *i* is served by driver *r* whose every day working zone is *z*. Coefficient u_{zr} is introduced to differentiate levels of inconsistent service. Higher value of u_{zr} represents larger deviation of consistent service. Namely, because driver's familiarity with a zone is a consequence of numerous factors (for example working experience in zone, living in zone, frequent visits to zone etc.) measure of service inconsistency should not be the same for all drivers. Therefore, we suppose that in the case when driver *r* serving zone *z* every day, i.e. when $r=z$, coefficient u_{zr} takes value of zero. In other cases u_{zr} takes value according to the decrease of service that driver *r* offers in zone *z*.

Constraints (2)-(9) are identical to constraints from (Toth and Vigo, 2002). Thus, constraint sets (2) and (3) impose that every task must be served by exactly one driver (vehicle) and that *k*+1 vehicle must leave depot, respectivley. Constraint sets (4) and (5) take care about the service sequence such that each task must have its predecessor (4) and successor (5). Notice that because only delivery part of distribution process is considered return of vehicles to the depot is not subject of the problem. Therefore, costs of transfer from any task to the depot are equal to zero, that is $c_{j0} = 0, \forall j \in V$. Constraints (6) respect vehicles' capacities, while subtour elimination constraints are given with (7).

Based on proposed model it is obvious that the CDMRS is generalization of the VRP. Because VRP are NP-hard, and due to the fact that networks served by PDC may include thousands tasks, it is clear that solving real size problems requires implementation of some heuristic algorithm. Therefore, in the following section we propose heuristic methodology to solving CDMRS.

It should be pointed out that by implementing different aggregating and grouping methodologies dimensions of the problem could be significantly reduced in terms of number of locations to be routed. However, in this paper we focus only on methodology for achieving maximal consistency of service with minimal routes' distance.

4 HEURISTIC APPROACH TO SOLVING CDMRS

Because of NP-hard nature of VRP, practical implementation of CDMRS implies use of a heuristic algorithm. We propose heuristic approach based on application of the following four steps:

- Step 1: determination of the number of delivery tasks vehicles serve within zones (including sweep vehicle)
- Step 2: simultaneous determination of specific delivery tasks to be served by non-consistent driver and routes of delivery vehicles
- Step 3: improvement of delivery vehicles' routes by local search algorithm
- Step 4: implementation of variable neighborhood descent (VND) algorithm to the solution found in step 3

In the first step by the values of decision variable v_{zr}, we define number of tasks that vehicle r serves within zone z, while assignment of specific tasks to be served by vehicle r is made in step 2. Because implementation of CDMRS requires minimization of driver to customer inconsistency mathematical model used for determination of v_{zr} is given by expressions (10) – (13).

$$\min \sum_{z \in Z} \sum_{r \in R} u_{zr} v_{zr} \tag{10}$$

Subject to

$$\sum_{r \in R} v_{zr} = \sum_{i \in V} b_{iz} \qquad \forall z \in Z \tag{11}$$

$$\sum_{z \in Z} v_{zr} \le C_r \qquad \forall r \in R \tag{12}$$

$$v_{zr} \in N \cup \{0\} \qquad \forall z \in Z, \forall r \in R \tag{13}$$

Objective function (10) minimizes service inconsistency measure. Constraint set (11) provides that all tasks from a zone z are served by some vehicle, while constraints (12) take care about vehicles' capacities. Constraints (13) allow variable v_{zr} to take value only from the set of natural numbers, including zero.

Task to vehicle allocation is determined in step 2. Step 2 of proposed heuristic approach includes not only final task to vehicle assignments but also simultaneous determination of tasks visiting sequence for each vehicle. In order to achieve this, based on the value of v_{zr}, vehicles (including sweep vehicle serving excess demand) are separated in four groups:

Group 1 (G1) – contains all vehicles that serve only tasks from consistent zone. Beside that, tasks from this zone are not served by any other (inconsistent) vehicle

Group 2 (G2) – contains all vehicles that beside tasks from consistent zone serve tasks in one or more inconsistent zones. However, tasks from the vehicle's consistent zone are not served by any other vehicle

Group 3 (G3) – contains all vehicles that serve only tasks from its consistent zones, but some of tasks from vehicle's consistent zones, because of excess demand, are served by the other zone inconsistent vehicles,

Group 4 (G4) – contains all vehicles that beside tasks from its consistent zones serve tasks in one or more inconsistent zones, while simultaneously tasks from vehicle's consistent zone are also served by vehicles inconsistent to the zone.

Defining task servicing sequence, for every vehicle *r,* includes consideration of all tasks that should be served within the both, vehicles' consistent and inconsistent zones. Therefore, the least effort is needed to determine sequence of task service for vehicles in G1, and the most efforts is required for sequencing tasks served by vehicles in G4. More precisely, in the case of G1, only tasks within vehicle's consistent zone are considered, while in the case of G4 first it should be decide which tasks from inconsistent zones will be served by certain vehicle, and which tasks from vehicle's consistent zone will be served by other vehicles inconsistent to that zone.

Therefore, due to the lack of allocation part of the problem, after the grouping of vehicles we continue step 2 by determining task service sequences (routes) for every vehicle *r* in G1. Routes are defined by applying insertion heuristics with objective to minimize route's length on a set of tasks from zone *z*. With each realization of the insertion algorithm considered vehicle is removed from G1. The insertion algorithm is executed until there are vehicles in G1.

Because of the characteristics that grouped vehicles in G2 it is obvious that for each vehicle *r* in G2 stands that routing algorithm is applied on a set of tasks consisting of all tasks from consistent zone *z* and $\sum_{h \in H} v_{hr}$ tasks from inconsistent zones. Set *H*, $H \subseteq Z$, contains all inconsistent zones in which vehicle *r* makes deliveries.

Usually vehicle *r* serves only a part of tasks in inconsistent zone *h*, meaning it is necessary to determine which tasks will be served by vehicle *r*. Therefore, by adding every possible combination of v_{hr} inconsistent zone tasks to the set of consistent zone tasks we get set of tasks on which insertion heuristic is implemented. Decision of final assignment of v_{hr} inconsistent zone tasks to vehicle *r* is made according to minimal length of generated routes. In cases when vehicle *r* delivers tasks in more then one inconsistent zone, i.e. when $|H|>1$, procedure for assigning tasks to vehicle *r* is repeated for each $h \in H$, respecting all tasks previously assigned to the vehicle *r*.

After all assignment procedures for zone *h*'s tasks are executed, remaining tasks from zone *h* are served only by consistent vehicle. Therefore, if there are no inconsistent tasks to be assigned to zone *h*'s vehicle, membership of zone *h*'s vehicles is updated from G3 to G1. Similarly, if beside tasks form zone *h,* zone *h*'s vehicle serves tasks in one or more inconsistent zones, membership of zone *h*'s vehicle is changed from G4 to G2. For that reason, after each execution of task assignment procedure, status of zone *h*'s vehicle is updated and if conditions are satisfied zone *h*'s vehicle is moved from G3 to G1 or from G4 to G2.

Eventually, after all inconsistent zones from vehicle *r*'s set *H* are considered, vehicle *r* is removed from G2 and described procedure for handling vehicles from G1 and G2 is repeated until G1 and G2 are empty sets. Diagram of proposed methodology for CDMRS is presented on Fig. 2.

Finally, in steps 3, and 4, we try to improve solution defined in step 2 by implementing two deterministic local search improvement algorithms. The first one is implemented in step 3. Algorithm takes step 2 solution as initial, and tries to improve it by using Or-opt moves with one, two and three tasks in every route. Precisely, one, two or three tasks are taken from existing route and inserted to every possible position within remaining part of the route. As an example, Or-opt move with two tasks is presented in Figure 3a.

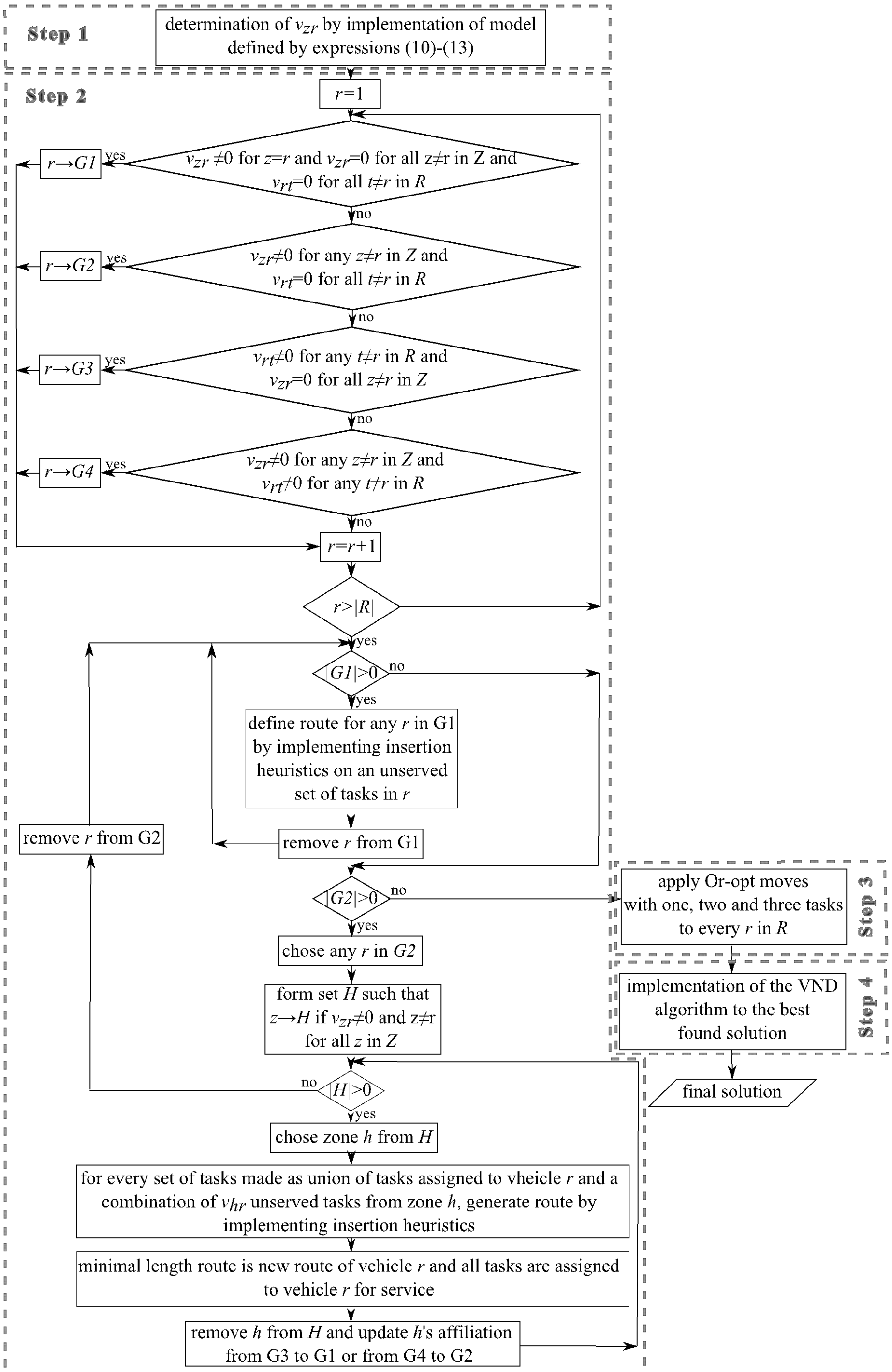


Figure 2. Overview of the heuristic algorithm for CDMRS

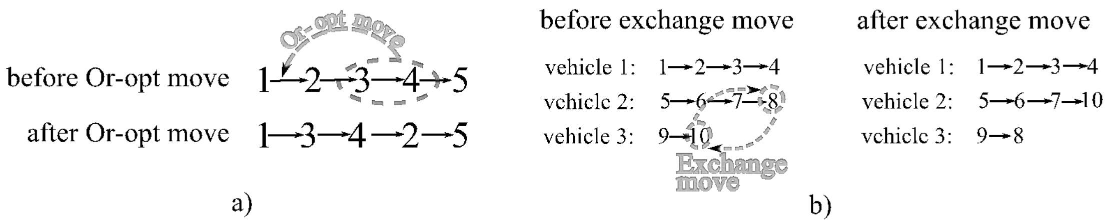


Figure 3. Moves used for generating neighborhood structures: Or-opt move (a), and exchange move (b)

Solution obtained in step 3 we further try to improve by changing solution's neighborhood structures in accordance with VND algorithm. VND algorithm is local search algorithm presented in Mladenovic and Hansen (1997) that tries to improve given solution by systematic exploration of solution's neighborhood structures. One of the most important decisions related with implementation of VND are neighborhood structures to be included in search. In this research VND includes two nested neighborhood structures. The first structure we consider consists of solutions gained by executing Or-opt moves from step 3 on existing solution. The second structure considered in this research consists of solutions obtained by exchange move on the best found solution. Exchange move implies that one task from a route is exchanged with all tasks on other routes (Figure 3b). However, in case of CDMRS such move is allowed only if capacity constraints on both routes are preserved and if consistency measure is not reduced. Because consistency and capacity constraints are not influenced by sequence of task service, and on the other side route length is very sensitive to service sequence, in proposed VND we implemented nested structures concept. This means that all solutions from the first neighborhood structure will be considered for every possible solution from the second neighborhood structure.

5 NUMERICAL EXAMPLES

Efficiency of proposed heuristic algorithm is compared to TSS on different instances of test examples which are made under the assumption that the number of stops during the day is the limiting factor in the parcels distribution process. Test instances have the following characteristics:

- Delivery area is represented as a square separated in a number of fixed delivery zones. Cases of four and sixteen delivery zones are considered.
- Delivery tasks are generated randomly over the delivery area with Euclidian distances between them.
- Due to a memory usage issues in solving the problem instances with large number of tasks to optimality, four zone problems include only 14 tasks, while sizes of the sixteen zone problems are uniformly distributed between 161 and 170 tasks.
- Capacity of each delivery vehicle in case of four zone instances is three stops and in the sixteen zone instances is ten stops. Number of regular vehicles is equal to the number of delivery zones.
- Two locations of depot within the delivery area (Fig. 4) are considered. In the first variant (V1) depot is located in the center of the delivery area, while in the second variant (V2) it is located in the center of one delivery zone.

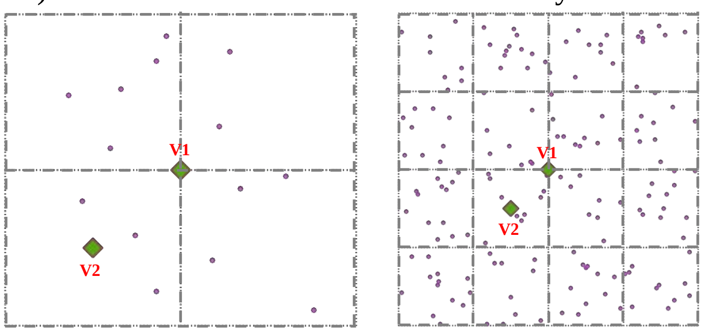


Figure 4. Example of test instances with four and sixteen delivery area

- Value of inconsistency parameter, u_{zr},, is supposed to be equal to the number of zone borders driver of the vehicle *r* must pass in order to reach zone *z*. Only orthogonal border passing is allowed.
- Value of u_{zr} for sweep vehicle is determined in the same way but with the difference that it is always increased by one. In this way consistent drivers are always preferred for service in their zones. Beside that, if depot is located on the border of zones we supposed that driver of sweep vehicle is equally familiar with adjacent zones and therefore inconsistency parameter has the same value for both zones.

For each combination of depot locations and delivery area structure we generated 100 instances. All algorithm runs are made on Win XP OS run by Intel Celeron 1.6GHz with 1GB of RAM. Optimal solutions are obtained by use of IBM ILOG CPLEX 12.2 MILP solver while all other by coding implemented in Python 2.5.

Summarized data of obtained results are given in tables 1 and 2. Table 1 contains data regarding both variants of four zone delivery area, while Table 2 contains data of both variants of sixteen zone delivery area. Information about average value and standard deviation of relative difference between an algorithm's solution and the best obtained solution are shown in "obj" columns of tables. Time requirements of implemented algorithms are shown in time columns of tables.

Table 1. Test results for the case of four zone delivery area

		Optimum		TSS		CDMRS-3		CDMRS-4	
		obj	time [s]	obj	time [s]	obj	time [s]	obj	time [s]
V1	average	0,00%	65,48	11,13%	>0.001	4,33%	0,003	0,99%	0,18
	st. dev.	0,00%	22,87	9,71%	>0.001	4,87%	0,002	2,54%	0,09
V2	average	0,00%	105,16	13,59%	>0.001	9,47%	0,003	3,71%	0,15
	st. dev.	0,00%	45,82	8,18%	>0.001	7,58%	0,002	4,22%	0,16

"Optimum" part of tables holds information regarding optimal solutions of instances. "TSS" section of tables contains information on solutions obtained by implementing TSS strategy, i.e. when only sweep vehicle is allowed to make inconsistent deliveries by serving excess of tasks (Fig. 1a). "CDMRS-3" and "CDMRS-4" parts of tables contain information regarding solutions obtained after third and fourth step of CDMRS , respectively.

Table 2. Test results for the case of sixteen zone delivery area

		TSS		CDMRS-3		CDMRS-4	
		obj	time [s]	obj	time [s]	obj	time [s]
V1	average	13,54%	0,017	3,71%	1,27	0,00%	38,14
	st. dev.	3,14%	0,006	1,63%	0,82	0,00%	9,32
V2	average	12,23%	0,017	3,31%	1,46	0,00%	36,36
	st. dev.	3,96%	0,003	1,71%	0,75	0,00%	10,94

Results from tables show reduction of route lengths in cases when instead sweep vehicle CDMRS is implemented. Savings obtained by CDMRS are around 10% and 13% in cases of four and sixteen zone delivery area, respectively. Also, from tables 1 and 2 it can be seen that implementation of VND algorithm reduces route lengths for approximately 3.5% in case of sixteen zones delivery area and for around 4.5% on average in the case of four zone delivery area.

From results in Table 1 it can be concluded that location of depot have significant influence on solution quality and problem complexity. However, this result is consequence of existence of small number of tasks and only four zones. Therefore, when depot is located in the center of delivery area, according to assumptions related to inconsistency measure of sweep vehicle, whole delivery area will be treated with the same value of inconsistency parameter. In such way, it is easier to find better solutions with equal level of consistency service. Confirmation of this claim are results in Table 2 from which it can be seen that there are no significant differences in algorithm efficiency for V1 and V2.

6 CONCLUSION

Significant time savings, provided by implementation of a fixed zone distribution strategy, imply to parcel delivery companies to restrict frequent changes in distribution zones. On the other side, in order to be able to efficiently respond to stochastic nature of delivery tasks PDCs have to implement some concept of route flexibility. In this research we considered situation in which due to stochastic nature of delivery tasks available capacity of vehicles is not sufficient to provide required quality of service. For this situation we proposed simple and easy applicable CDMRS strategy that, compared to strategy in which sweep vehicle is used to serve excess tasks over delivery area, prove to reduce overall route length while minimizing the level of inconsistent deliveries.

Because number of deliveries in a PDC’s depot is measured in thousands, in this research we supposed that reduction of problem size is realized prior to the implementation of CDMRS strategy. However, since practical application of proposed strategy implies execution of problem size reduction procedure, it is clear that one direction for further research regarding this problem is development of such procedure. The other direction for further research is related to improvement of proposed approach through implementation of more efficient improvement algorithms.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education and Science of the Government of the Republic of Serbia through the projects TR36005 and TR36006, for the period 2011-2014.

REFERENCES

Mladenovic N., Hansen P. (1997) Variable neighborhood search, *Computers and Operations Research*, **24/11**, 1097-1100

Sorensen, K., Van den Berg, J., Cattrysse D., Sevaux M. (2008), A multi-objective distribution problem for parcel delivery at TNT, *Talk at International Workshop on Vehicle Routing in Practice 2008*, Oslo, Norway, June 12-14, 2008.

Schneider M., Stenger A., Lagemann H., Vigo D., (2012), On Fixed-Area-Based Vehicle Routing in the Presence of Time Window Constraints, *Tech. Report 01/2012;BISOR*, TU Kaiserslautern

Toth, P. and Vigo D. (2002), An Overiew of Vehicle Routing Problems, In P. Toth and D. Vigo (eds), *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia

Wong R.T. (2008), Vehicle Routing for Small Package Delivery and Pickup Services, In B. Golden, S.Raghavan, E.Wasil (eds), *The Vehicle Routing Problem: Latest Advances and New Chalanges*, Springer Science+Business Media, New York

Zhong, H., Hall, R.W., Dessouky, M. (2007), Territory Planning and Vehicle Dispatching with Driver Learning, *Transportation Science*, **47/1**, 74-89

A STUDY ON THE DESIGN AND PERFORMANCE OF A ZONE-PICKING SYSTEM WITH COOPERATION AREA BETWEEN NEIGHBORING ZONES

Ying-Chin Ho[1] and Jia-Chyun Chen[2]

[1]*Institute of Industrial Management, National Central University, Chung-Li, 32001, Taiwan, ho@cc.ncu.edu.tw*

[2]*Institute of Industrial Management, National Central University, Chung-Li, 32001, Taiwan, 944406002@cc.ncu.edu.tw*

ABSTRACT

In a zone-picking system, pickers cannot enter zones that they do not belong to. As pickers are performing their picking tasks, some pickers may be busier than the others. In other words, the workload between zones is not balanced. The unbalanced workload between zones can hinder the performance of the entire system. Furthermore, the picking time span cannot really be minimized because of the no-zone-crossing restriction. In this paper, we propose a new zone-picking system that has cooperation area set up between neighboring zones. Pickers can enter the cooperation area of their neighbouring zones to perform the cooperation task. The problems associated with this new system include the item clustering problem, the problem of assigning item groups to zones, the storage assignment problem, the cooperation area design problem and the cooperation-strategy problem. Various methods will be developed for the above problems. Simulation experiments will be conducted to test the performance of these methods and to show that setting cooperation area between neighbouring zones is beneficial to our zone-picking systems.

Keywords: warehouse, zone picking, cooperation area

1 INTRODUCTION

Nowadays, one often refers to a warehouse as a distribution centre if its distribution function is more important than its storage function. In warehouses or distribution centres, products have to be picked from specified storage locations on the basis of customer orders. According to Gross (1981), ordering picking operations account for about 30-40% of the labour cost in a traditional warehousing system. One expects the percentage to be even greater if the warehousing system is a distribution centre in which all of its order-picking operations are performed manually. Coyle *et al.* (1996) also pointed out that order-picking operations account for about 65% of the total operating cost in a typical warehouse. Thus, in order to minimize the operation cost of a distribution centre, one must optimize its order-picking operations. Previous studies have come up with various methods to improve order-picking operations, such as the automation and computerization of the warehousing system, order batching, zone picking, class-based storage, storage assignment policies, routing methods, layout design, etc. In this study, we concentrate on improving the efficiency of a zone picking system.

In a zone-picking system, the order picking area is divided into zones. Each order picker performs the order picking tasks in his/her zone. The advantages of zone picking include a smaller travel distance for each order picker, reduced traffic congestion, and the possibility that order pickers become familiar with the item locations in the zone. Generally, there are two approaches of zone picking. The first approach is the progressive assembly of an order.

Using this approach, when an order picker has finished the picking task of an order in his/her zone, he/she will hand over the picking box (tote) and pick list to the order picker in the next zone. The next order picker will continue the assembly of the order. Hence an order is only finished after it has visited all relevant zones. This system is called pick-and-pass system, which is also known as the sequential zone-picking system. The sequential zone-picking system is not the zone-picking system we focus on here. The zone-picking system we have here is modified from the system that adopts the second approach – parallel picking. This system is also known as the synchronized zone-picking system. In this approach, an order can be divided into several sub-orders. A number of order pickers can start working on sub-orders that belong to the same order at the same time. And, in the traditional synchronized zone-picking system, each order picker can only work in the zone he/she is assigned to. In other words, an order picker cannot leave the zone he/she is assigned to. The sub-orders belonging to the same order are merged afterwards at the order-consolidation area. In our proposed synchronized zone-picking system, we assume that the warehouse has a radio frequency identification (RFID) system that allows us to receive the real-time information of every picking cart's location, and the status of all orders and picking zones. We also allow order pickers to leave their zones and enter the cooperation area of their neighbouring zones. A more detailed description of our proposed synchronized zone picking system will be presented in Section 3.

In this paper, we propose a new zone-picking system that has cooperation area set up between neighboring zones. Pickers can enter the cooperation area of their zones to perform the cooperation task. The problems associated with this new system include the item clustering problem, the problem of assigning item groups to zones, the storage assignment problem, the cooperation area design problem and the cooperation-strategy problem. The objective of this paper is to develop practical methods for the above problems and to show that the design with cooperation area between neighboring zones can indeed outperform the traditional zone-picking system without cooperation area between neighboring zones.

2. LITERATURE REVIEW

Order picking has been studied by many researchers. A great number of studies have focused on increasing the efficiency of order picking, such as storage assignment and routing policies. In storage assignment policies, Hausman *et al.* (1976) examined the performance of an automated warehouse with random and class-based storage. They found class-based storage assignment policies could reduce the crane travel times in an automated warehousing system significantly. Jarvis and McDowell (1991) stated that the optimal storage strategy is to place the most frequently picked items in the aisle nearest the P/D (Pickup/Delivery) point and the next most frequently picked items in the next aisle. Their research was limited in several places, e.g. uni-directional travel is allowed in aisles and only transversal routing is allowed.

Based on simulation experimental results, Petersen *et al.* (2004) showed that in terms of the travel distance in a manual order-picking system, full-turnover storage outperforms class-based storage. The gap between them depends on the class partition strategy (i.e. number of classes, percentage of the total volume per class) and the routing method used. However, they suggest using the class-based method with 2 to 4 classes in practice, as it is easier to implement than the volume-based method. Furthermore, it does not require a complete list of the items ranked by volume and it requires less time to administer than the other dedicated methods.

As for routing policies, Ratliff and Rosenthal (1983) developed an algorithm for finding the shortest order-picking route. Hall (1993) developed distance approximation methods for five order-picking strategies – traversal strategy, midpoint strategy, largest-gap strategy, optimal routing, and minimal strategy. Petersen (1997) investigated the performance of six heuristic routing strategies and compared them to the optimal strategy. Roodbergen and De Koster (2001) investigated the problem of route-planning problems in a warehouse with multiple cross-aisles. They considered several heuristics to determine order-picking routes. To analyze the performance of their heuristics, a branch-and-bound algorithm that generates shortest order-picking routes was used.

Some researchers adopted the zone-picking method in which every picker only picks those items that are in the zone assigned to them. For example, De Koster (1994) modeled a zoned pick-and-pass system as a Jackson queuing network, which allows rapid estimations of order throughput times and average work-in-process. Results are compared with simulations. The estimates can be used to determine the number of zones and the system size. Mellema and Smith (1988) examined the effects of the aisle configuration, stocking policy and batching and zoning rules by using simulation. They suggested that a combination of batching and zoning can significantly increase the productivity. Also, using simulation, Petersen (2002) showed that the zone shape, the number of items on the pick-list and the storage policy have a significant effect on the average travel distance within the zone.

Choe *et al.* (1993) studied the effects of three strategies – single-order-pick, sort-while-pick, and pick-and-sort – in an aisle-based order picking system. They propose analytical tools for a planner to quickly evaluate various alternatives without using simulation. Jane (2000) proposed several heuristic algorithms to balance the workloads among the order pickers and to adjust the zone size for order volume fluctuation in a progressive zoning system. Jane and Laih (2005) considered the problem of heuristically assigning products to zones in a synchronized system. The method is based on co-appearance of items in the same order, i.e. items appear in the same order are stored in the same zone. Jewkes *et al.* (2004) tackled the product assignment problem, zone sizing, and picker home base location for a progressive system. Their method is based on dynamic programming. Using a mixed-integer linear programming model, Le-Duc and De Koster (2005) determined the optimal number of zones in a synchronized zoning system so that the total order-picking and assembly time is minimized.

Koo (2009) discussed distinct characteristics in order picking systems when bucket brigades (BB) are applied. He identified some efficiency losses under the BB picking and presented a new BB picking protocol to improve the performance of order picking systems. Eisenstein (2008) considered three levels of discrete order-picking technology along a line: (1) no technology (single-depot configuration); (2) conveyor technology (dual-depot configuration); (3) conveyor and RF technology (no-depot configuration). He then built a stochastic model to compare these three configurations of different technology requirements and to explore the optimal design for each configuration.

Parikh and Meller (2008) focused on the problem of selecting between a batch picking and a zone picking strategy. They proposed a cost model to estimate the cost of each type of picking strategy. In the cost model, they considered the effects of pick rate, picker blocking, workload imbalance and the sorting system requirement. Gong (2009) studied stochastic models and analysis of warehouse operations. He built models to describe and analyze such systems via stochastic polling theory, found closed-form expressions for the order line waiting times, and applied polling-based picking to online retailers. De Koster *et al.* (2007) gave a literature overview on typical decision problems in the design and control of manual order-picking processes. They focused on optimal (internal) layout design, storage assignment methods, routing methods, order batching and zoning.

RFID-based technologies have been widely applied in numerous areas in the supply chain activities. Chow *et al.* (2006) designed an RFID case-based resource management system for warehouse operations, and presented how RFID-RMS could be used in improving the warehouse operating performance. Chen *et al.* (2009) applied RFID technologies to picking operation in warehouses, and presented a complete process for order picking system with the help of RFID technology.

3 THE PROBLEM ENVIRONMENT

3.1 The Traditional Zone-Picking Warehouse

In this study, a traditional zone-picking warehouse is divided into several picking zones (see Figure 1). Each picking zone has its own P/D point. P/D points are the places that pickers pick up their picking list in a picking trip. They are also the place where human pickers return to deposit the items they have picked on the conveyor. Pickers can only work in the picking zones they belong to. In this zone-picking system, an order may have to be divided into a number of different sub-orders. Each sub-order belongs to a picking zone. The maximum number of sub-orders that an order can be divided into is equal to the number of zones in the zone-picking system.

A picking trip begins when a picker arrives at the P/D point and enters his/her personal code to the computer. The computer gives the picker an instruction telling him/her the list of items (i.e., an order) he/she needs to pick. There are many picking blocks in a picking zone. When the picker arrives at a block, he/she can enter the block from either end of the block since each end of a block is equipped with an RFID reader. The reader reads the RFID chip in the picker's wrist band. Once a picker enters a picking block, he/she is assisted by a pick-to-light system in performing his/her picking operations. After a picker has completed his/her picking task in a picking block, he/she can leave the block from either end. Please note again the zone-picking system we have here is not a sequential zone-picking system. In a sequential zone-picking system, zones are arranged sequentially. A picking box (tote) must pass through every zone. However, in our zone-picking system, picking boxes (totes) cannot be passed between different zones. In other words, once a picking box (tote) has had all of the items it requires, it will be placed on the conveyor which delivers it to the order consolidation zone (see Figure 1).

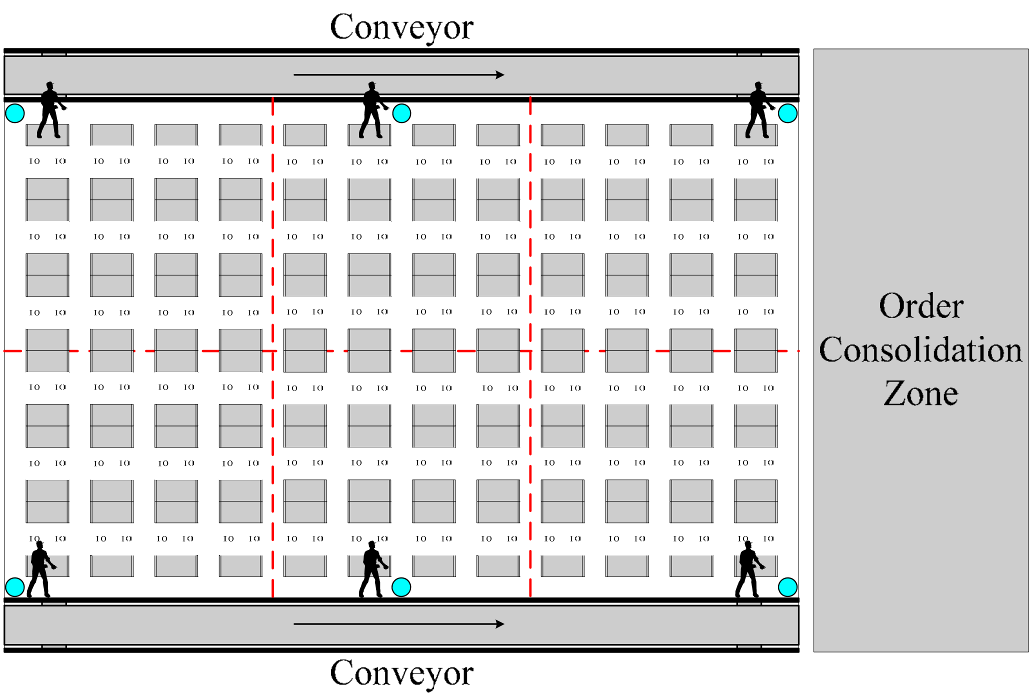


Figure 1. An Example of a Traditional Zone-Picking Warehouse

3.2 The Proposed Warehouse Environment – A Zone-Picking Warehouse with Cooperation Area

In this study, we proposed a new warehouse environment. This warehouse environment is similar to the traditional warehouse environment introduced above. The main difference between them is that a cooperation area is set up between neighbouring zones. A picker of a zone, e.g. *Zone A*, can enter the cooperation area in a neighboring zone of *Zone A*. We also assume that the proposed warehouse environment has an RFID system that allows us to receive the real-time information of every picking cart's location, and the status of all orders and picking zones.

4 THE PROPOSED METHODS

In the following, we will discuss several problems that are associated with a warehouse with cooperation area. They include the item clustering problem, the problem of assigning item groups to zones, the storage assignment problem, the cooperation area design problem and the cooperation-strategy problem. These problems and the methods proposed for them are presented in the following sections.

4.1 The Proposed Method for the Item Clustering Problem

In the clustering problem, we divide items (i.e. stock keeping units) into groups. Items that are clustered into the same group will be placed in the same zone. For this problem, a two-phase method is proposed.

4.1.1 The First Phase of the Item Clustering Problem

At the fist phase, we calculate the similarity coefficients between items. Three similarity coefficient measures are studied here. They are Ochiai (OSC), Baroni-Urbani and Buser (BSC), and Sorenson (SSC). They are defined as follows.

$$OSC_{ij} = \frac{a}{\sqrt{(a+b)(a+c)}} \tag{1}$$

$$BSC_{ij} = \frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}} \tag{2}$$

$$SSC_{ij} = \frac{2a}{2a+b+c} \tag{3}$$

where,

a = the number of orders that need items i and j,
b = the number of orders that need item i only,
c = the number of orders that need item j only, and
d = the number of orders that need neither i nor j.

4.1.2 The Second Phase of the Item Clustering Problem

At the second phase, items are clustered into different item groups. Three clustering methods are studied here. They are random, cumulative, and fuzzy c-means. In the random clustering method, items are randomly clustered into groups. In the cumulative clustering method, seed items are first selected for item groups. Each item group is assign with a seed item. After that, items are added into each group based on their similarity coefficients with the items that have been added into the group. The adding procedure is repeated until all items have all been added into item groups. In the fuzzy-c means clustering method, the fuzzy-c means approach is used to cluster items into item groups.

4.2 The Proposed Method for Assigning Item Groups to Zones

For the problem of assigning item groups to zones, we propose a simple yet efficient method by first assigning each zone with an item group. After that, a pair-wise exchange method is adopted to further improve the assignment result. The goal is to find an assignment result in which two item groups with a great amount of identical orders can be assigned to zones that are close to each other.

4.3 The Proposed Method for the Storage Assignment Problem

After the problem of assigning item groups to zones has been solved, the next problem for us is the storage assignment problem in each picking zone. In this paper, we use four different methods. The first two methods are volume-based methods. We refer to them as the vertical-assignment method and the horizontal-assignment method. In both methods, an item with a greater demand volume will be assigned to a place that is closer to the P/D point. The difference between these two methods is their assignment patterns – one is vertical and the other is horizontal. Figure 2 illustrates these two methods.

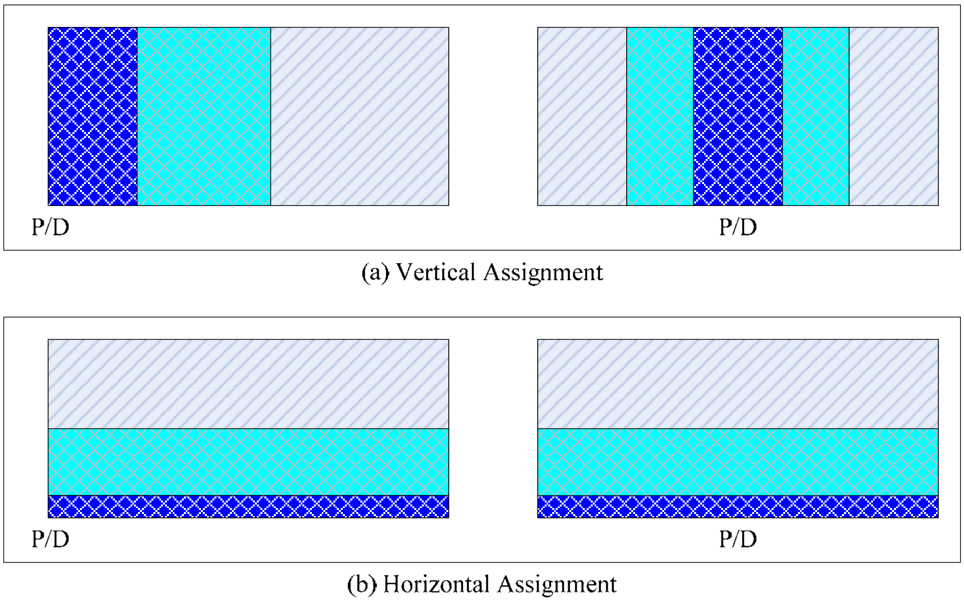


Figure 2. Two Volume-based Aassignment Methods

We refer to the third method as the center-of-order-gravity method. With this method, we use an equation (see Eq. (4)) to obtain center-of-order-gravity location of an item. We then use this location to find the best place that this item should be assigned to. Finally, the last storage assignment method is the random assignment method which randomly assigns items to their storage locations. The purpose of the random assignment method is for the comparison purpose. It is apparent a storage assignment method is not worth adopting if its performance is worse than that of the random assignment method.

$$(X_i, Y_i) = \left(\frac{\sum_{c=1}^{N} (ZO_{ci} \times x_c)}{\sum_{c=1}^{N} ZO_{ci}}, \frac{\sum_{c=1}^{N} (ZO_{ci} \times y_c)}{\sum_{c=1}^{N} ZO_{ci}} \right) \text{, for } c = 1、2\ldots N \text{ \& } i = 1、2\ldots m. \quad (4)$$

where,

c = the zone index,
N = the number of zones,
ZO_{ci}= the number of orders which include item i in zone c,
(x_c, y_c) = the coordinates of the P/D point in zone c,
(X_i, Y_i) = the coordinates of the centre-of-order-gravity of item i, and
m = the total number of items.

4.4 The Proposed Methods for the Cooperation Area Design Problem

In this study, we suggest four design methods for setting up the cooperation area between zones. They include Vertical, Horizontal, Hybrid and Diagonal. The patterns of these design methods are illustrated in Figure 3. Please note the Hybrid pattern is the combination of the Vertical Pattern and the Horizontal Pattern. These methods will be compared with each other in the simulation experiments.

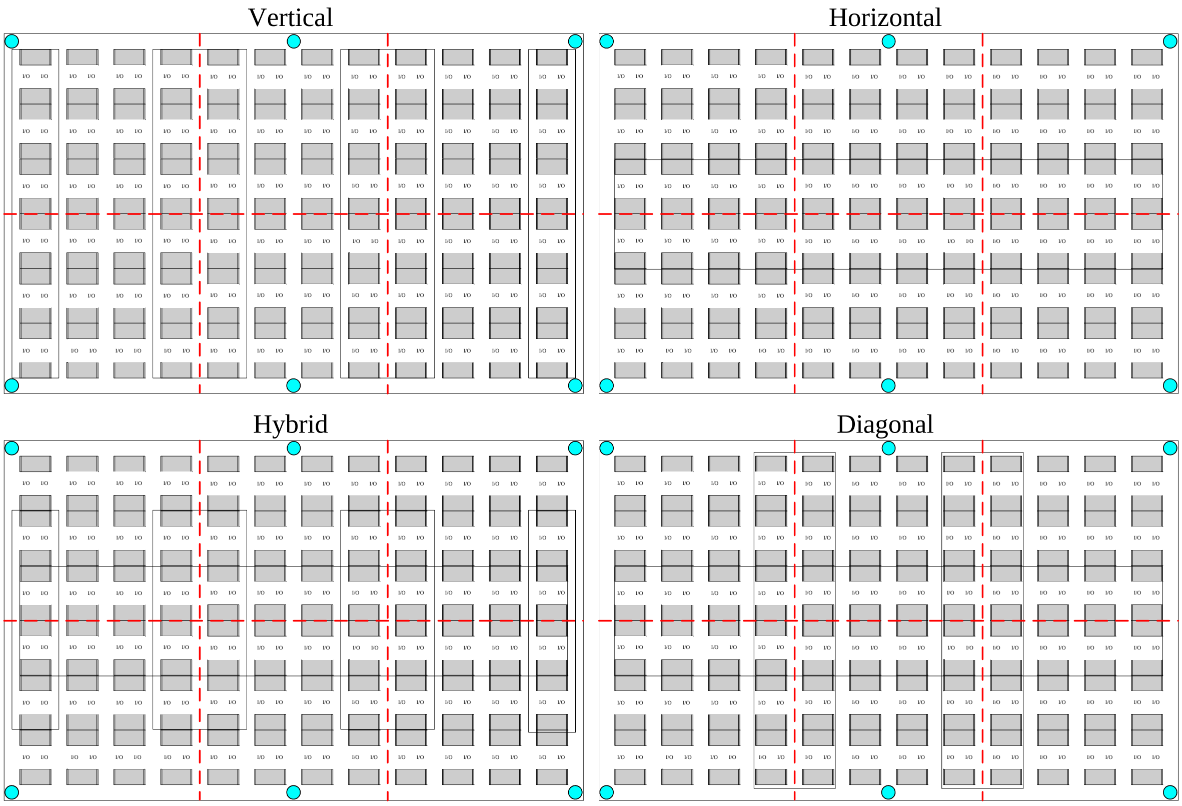


Figure 3. Designs of the Cooperation Area between Zones

3.7 Cooperation Strategies

A cooperation strategy tells pickers of neighboring zones when and how to help each other. Six cooperation strategies are studied in this paper. They are listed and described as follows.

- No Cooperation (NC):
  Under this strategy, human pickers are not allowed to help each other as in the traditional zone-picking warehouse. This strategy is included in the study for the purpose of comparison. A cooperation strategy is not worse adopting if it performs worse than the no-cooperation strategy.
- Always Help (AH):

We give an example to explain this strategy. A human picker, say *H*, in a zone, say *Z1*, starts picking items from a sub-order, say *SO*(*Z1*) which comes from another order, say *O*. Zone *Z2* is the neighbouring zone of Zone *Z1*. Another sub-order *SO*(*Z2*) of order *O* is currently waiting to be picked by human pickers in Zone *Z2*. And, some items, say *I*, *J*, and *K*, of sub-order *SO*(*Z2*) are located at the cooperation area in Zone *Z2*. Under this situation, the human picker *H* will go to the cooperation area in Zone *Z2* to pick items *I*, *J*, and *K*.

- Picking-Time-Ratio-based Strategy (PTRS):

  We also give an example to explain this strategy. A human picker, say *H*, in a zone, say *Z1*, starts picking items from a sub-order, say *SO*(*Z1*), which comes from another order, say *O*. Zone *Z2* and Zone Z3 are the neighbouring zones of Zone *Z1*. They both have sub-order *SO*(*Z2*) and *SO*(*Z3*) that also come from order *O*. The picker in Zone Z2 and Zone Z3 are currently busy on picking items of other orders. We use Equation (5) to calculate the picking-time ratio (PTR) between Z2 and Z1 and the picking-time ratio between Z3 and Z1. The human picker will go to the cooperation zone in Z2 or Z3 or both to pick the items belonging to *O* depending on if their picking-time ratios are greater than a threshold value that is pre-defined by users. In this study, two PTR threshold values, 1.5 (T1) and 2.0 (T2), are tested.

$$PTR_{ji} = \frac{(NWT_j + NTT_j)}{NTT_i} \quad \text{for } i \neq j \tag{5}$$

where,

NWT_j = the estimated remaining picking time for some sub-orders currently being picked in Zone *j*. These sub-orders do not belong to the same order of the sub-order that the human picker in Zone *i* has just started working on,

NTT_j = the estimated required picking time for the sub-order that will be being picked in Zone *j*. This sub-order belongs to the same order of the sub-order that the human picker in Zone *i* has just started working on, and

NTT_i = the estimated required picking time for the sub-order that the human picker in Zone *i* has just started working on.

- Number-of-Picking-Items-Ratio-based Strategy (NPIRS):

  This strategy is similar to PTRS except it is based on the number of picking items ratio. For example, using the same example, the number of picking items ratio (NPIR) between Z2 and Z1 and the NPIR between Z3 and Z1 will be calculated and compared with the NPIR threshold value given by users. In this study, two NPIR threshold values, 1.5 (Q1) and 2.0 (Q2), are tested.

4 EXPERIMENTAL RESULTS

Simulation experiments were conducted to understand the performance of various methods described. For the simulation experiments, we randomly generated twenty order pools. In each order pool, there are one hundred randomly generated orders. Each order pool was used for one simulation replication. In the simulations, the travel speeds of all order pickers (with pick carts) were assumed to be 30 meters per minute. Table 1 shows the factors considered in the experiments and their levels. There are 1080 (3×3×4×5×6) combinations of different levels of factors. For each combination, twenty simulation replications were conducted using twenty randomly generated order pools. Figure 4 shows the problem environment in the simulation experiments.

The performance measure is the total system time (TST) which is the amount of time for the warehouse system to complete all picking operations. In other words, it is the picking-time-span. Because of the page number limit on this paper, only the TST performance of each main factor’s different levels is presented below. The mutual effects between different main factors are not presented. A comprehensive discussion on the mutual effects of main factors will be presented in the subsequent journal publication of this conference paper.

Table 1. The Factors Considered in the Simulation Experiments and Their Levels

Factors	Similarity Coefficient	Similarity Coefficient	Storage Assignment Method	Cooperation-Area Design Method	Cooperation Strategy
Level	OSC	Fuzzy c-Means	Centre-of-Order-Gravity	Diagonal	NPIRS-Q1
	SSC	Cumulative	Vertical-Assignment	Hybrid	PTRS-T1
	BSC	Random	Horizontal-Assignment	Horizontal	PTRS-T2
			Random-Assignment	Vertical	NPIRS-Q2
				No Cooperation Area	No Cooperation
					AH

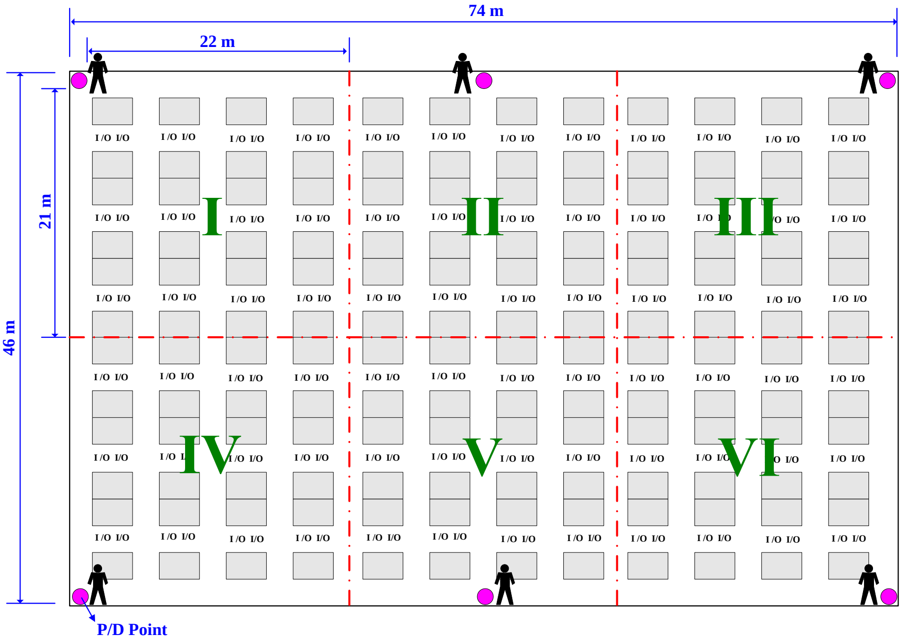


Figure 4. The Problem Environment in the Simulation Experiments

The ANOVA results indicate that main effects are significant at an α of 0.05. Table 2 shows the Duncan test result on the TST performance of similarity coefficients. As shown in Table 2, the OSC coefficient has the best performance, followed by SSC and BSC. Furthermore, the performance difference among them is significant at an α of 0.05.

Table 2. The Duncan Test Result on the TST Performance of Similarity Coefficients

Similarity Coefficient	Subset		
	1	2	3
OSC	168.04		
SSC		179.01	
BSC			183.59

Table 3 shows the Duncan test result on the TST performance of clustering methods. As shown in Table 3, the Fuzzy c-Means method is significantly better (at an α of 0.05) than the Cumulative method and the Random method. On the other hand, the Cumulative method is significant better (at an α of 0.05) than the Cumulative method and the Random method.

Table 3. The Duncan Test Result on the TST Performance of Clustering Methods

Clustering Method	Subset		
	1	2	3
Fuzzy c-Means	140.97		
Cumulative		173.76	
Random			215.92

Table 4 shows the Duncan test result on the TST performance of storage assignment methods. As shown in Table 4, all proposed methods (i.e. the Centre-of-Order-Gravity method, the Vertical Assignment method and the Horizontal Assignment method) are all significantly better (at an α of 0.05) than the Random Assignment method. Among them, the Centre-of-Order-Gravity method has best TST performance. Its performance is significantly better (at an α of 0.05) than that of the other methods.

Table 4. The Duncan Test Result on the TST Performance of Storage Assignment Methods

Storage Assignment Method	Subset			
	1	2	3	4
Centre-of-Order-Gravity	126.83			
Vertical-Assignment		157.78		
Horizontal-Assignment			168.24	
Random-Assignment				254.68

Table 5 shows the Duncan test result on the TST performance of cooperation-area design methods. As shown in Table 4, all proposed methods are all significantly better (at an α of 0.05) than the no-cooperation-area method. Among them, the Diagonal method has best TST performance. Its performance is significantly better (at an α of 0.05) than that of the other methods.

Table 5. The Duncan Test Result on the TST Performance of Cooperation-Area Design Methods

Cooperation-Area Design Method	Subset			
	1	2	3	4
Diagonal	174.4			
Hybrid		175.46		
Horizontal		175.62		
Vertical			176.87	
No Cooperation Area				182.05

Table 6 shows the Duncan test result on the TST performance of cooperation strategies. As one can see in Table 5, NPIRS-Q1 and PTRS-T1 strategies are in the first subset. This means they are significantly better (at an α of 0.05) than the other cooperation strategies. However, the difference between them is not significant (at an α of 0.05). From Table 6, we find the AH strategy is not better than the no-cooperation strategy. It seems that the non-selective cooperation approach of the AH strategy is not a wise approach.

Table 6. The Duncan Test Result on the TST Performance of Cooperation Strategies

Cooperation Strategy	Subset			
	1	2	3	4

NPIRS-Q1	172.33			
PTRS-T1	172.4			
PTRS-T2		174.55		
NPIRS-Q2			177.81	
No Cooperation				182.05
AH				182.14

5 CONCLUSION

In a typical zoning warehouse, each order picker can only pick items in the zone assigned to him/her. This typical environment may cause the unbalanced workload between zones and make the whole warehouse system inefficient. In order to avoid this inefficiency, a new zone-picking system that has the cooperation area set up between neighboring zones is proposed in this study. Pickers can enter the cooperation area near their zones to perform the cooperation task and balance the workload between zones. In addition, we propose several methods to solve the problems associated with this new system. These problems include the item clustering problem, the problem of assigning item groups to zones, the storage assignment problem, the cooperation area design problem and the cooperation-strategy problem. The results of the experiment prove that the new zone-picking system with cooperation areas can improve the efficiency of the warehouse system. Finally, it is hoped that knowledge learned from this study can benefit other researchers in developing better cooperation strategies and better design methods of the cooperation area in the future research.

REFERENCES

Chen, K. Y., Hwang, Y. F. and Chen, M. C. (2009), Applying RFID to picking operation in warehouses, In S.-Y. Chou, A. Trappey, J. Pokojski and S. Smith (eds), *Global Perspective for Competitive Enterprise, Economy and Ecology* (*Proceedings of the 16th ISPE International Conference on Concurrent Engineering*), Springer, London.

Choe, K. I., Sharp, G.P., Serfozo, R.S. (1993). Aisle-Based Order Pick Systems with Batching, Zoning and Sorting. In: R. J. Graves *et al.* (eds), *Progress in Material Handling Research: 1992*. The Material Handling Institute of America: Charlotte, NC, 245-276.

Chow, H. K. H., Choy, K. L., Lee, W. B. and Lau, K. C. (2006), Design of a RFID Case-based Resource Management System for Warehouse Operation, *Expert Systems with Applications*, **30(4)**, 561-576.

Coyle, J., Bardi, E.J., Langley, C. J., (1996), *The Management of Business Logistics*, 6^{th} Edition, West Publishing, St. Paul, MN.

De Koster, R. (1994), Performance Approximation of Pick-to-Belt Order Picking Systems, *European Journal of Operational Research*, **72(3)**, 558-573.

De Koster, R., Le-Duc, T. and Roodbergen, K. J. (2007), Design and Control of Warehouse Order Picking: A Literature Review, *European Journal of Operational Research*, **182(2)**, 481-501.

Eisenstein, D. D. (2008), Analysis and Optimal Design of Discrete Order Picking Technologies Along a Line, *Naval Research Logistics*, **55(4)**, 350-362.

Gong, Y. (2009), *Stochastic Modelling and Analysis of Warehouse Operations*, Ph.D. Thesis, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam, Netherland.

Gross, G. (1981), Picking Method May Provide Key to Lower Cost Warehouse Plans, *Industrial Engineering*, **13(6)**, 50-54.

Hall, R. W. (1993), Distance Approximations for Routing Manual Pickers in a Warehouse, *IIE Transactions*, **25(4)**, 76-87.

Hausman, W. H., Schwarz, L. B. and Graves, S. C. (1976), Optimal Storage Assignment in Automatic Warehousing Systems, *Management Science*, **22(6)**, 629-638.

Jane, C. C. (2000), Storage Location Assignment in a Distribution Centre, *International Journal of Physical and Logistics Management*, **30(1)**, 55-71.

Jane, C. C. and Laih, Y. W. (2005), A Clustering Algorithm for Item Assignment in a Synchronized Zone Order Picking System, *European Journal of Operational Research*, **166(2)**, 489-496.

Jarvis, J. M. and McDowell, E. D (1991), Optimal Product Layout in an Order Picking Warehouse, *IIE Transactions*, **23(1)**, 93-102.

Jewkes, E., Lee, C. and Vickson, R. (2004), Product Location, Allocation and Server Home Base Location for an Order Picking Line with Multiple Servers, *Computers and Operations Research*, **31(4)**, 623-636.

Koo, P. H. (2009), The Use of Bucket Brigades in Zone Order Picking Systems, *OR Spectrum*, **31(4)**, 759-774.

Le-Duc, T. and De Koster, R. (2005), Determining the Optimal Number of Zones in a Pick-and-Pack Order Picking System, Report ERS-2005-029-LIS, RSM Erasmus University, the Netherlands.

Mellema, P. M. and Smith, C. A. (1988), Simulation Analysis of Narrow-Aisle Order Selection Systems, *Proceedings of the 1988 Winter Simulation Conferenc*e, 12-14 December 1988, San Diego, CA., 597-602.

Parikh, P. J. and Meller, R. D. (2008), Selecting between Batch and Zone Order Picking Strategies in a Distribution Center, *Transportation Research Part E: Logistics and Transportation Review*, **44(5)**, 696-719.

Petersen, C. G. (1997), An Evaluation of Order Picking Routeing Policies, *International Journal of Operations and Production Management*, **17(11)**, 1098-1111.

Petersen, C. G. (2002), Considerations in Order Picking Zone Configuration, *International Journal of Operations and Production Management*, **22(7)**, 793-805.

Petersen, C. G., Aase, G. R., and Heiser, D. R. (2004), Improving Order-Picking Performance through the Implementation of Class-Based Storage, *International Journal of Physical Distribution and Logistics Management*, **34(7)**, 534-544.

Ratliff, H.D. and Rosenthal, A.S. (1983), Order-Picking in a Rectangular Warehouse: a Solvable Case of the Traveling Salesman Problem, *Operations Research*, **31(3)**, 507-521.

Roodbergen, K. J. and De Koster, R. (2001), Routing Methods for Warehouses with Multiple Cross Aisles, *International Journal of Production Research*, **39(9)**, 1865-1883.

ENDOSYMBIOTIC EVOLUTIONARY ALGORITHM FOR VEHICLE ROUTING PROBLEMS WITH DELIVERY AND INSTALLATION

Kyoung Cheol Kim[1], Ji Ung Sun[2] and Shi W. Lee[3]

[1]*School of Mechanical, Industrial & Manufacturing Engineering, Oregon State University, Corvallis, OR 97331, USA*

[2]*School of Industrial & Management Engineering, Hankuk University of Foreign Studies, Seoul, Korea*

[3]*Department of Industrial Engineering, Pusan National University, Busan, Korea*

ABSTRACT

Efficient and effective planning methods to provide visible and invisible goods are required to satisfy demands of business and customers in a competitive market. Vehicle routing problem (VRP) has been extensively studied for various issues with special needs, as one of important planning methods to reduce waists and maximize profits. In this paper, a VRP with unique characteristics of the electronic market industry is considered. The VRP deals with two types of customers' demands and two types of service vehicles, delivery and installation. In order to solve the VRP under consideration, an endosymbiotic evolutionary algorithm (EEA) is proposed to handle the two different services simultaneously and implemented with appropriate and efficient processes. The computational results show the performance of the proposed algorithm with a comparison among other approaches to the VRP under consideration.

1 INTRODUCTION

Businesses and industries in competitive environment require efficient planning methods to handle various visible and invisible products. Especially in the manufacturing industry, material handling methods are important for not only final products but also raw materials and works-in-process. As the consideration of vehicle routes and schedules for the cost-effective supply is one of issues in the material handling methods, vehicle routing problem (VRP) has been widely and extensively investigated for various industries with special needs such as for soft drink delivery (Golden and Wasil, 1987; Prive et al, 2006), hazard material transportation (List and Mirchandani, 1991; Zografos and Androutsopoulos, 2004), school bus routing (Chien and Spasovic, 2001; Ripplinger, 2005) and postal service (Ji, 2007). An industry dependent VRP needs distinctive characteristics.

In this paper, a VRP, characterized in the electronics market industry to satisfy needs from business and customers as the paradigm of distribution is shifted from the past, is considered. First, the electronics market industry has recently faced changes in its post-sales service such as delivery and installation. In the past, local electronics stores were individually required to directly delivery and install products to their customers. However, these days, large electronics vendors are increasingly required to directly deliver products to their customers and to provide on-site installation. Furthermore, various types of electronics sales via e-commerce, large discount stores, general merchandise stores, department stores, and professional electronic franchises are increasing very rapidly. These trends tend to push greater responsibilities for not only delivery but also installation onto current electronics vendors. Moreover, the number of direct deliveries from electronics vendors to customers continues to increase as well. Second, many electronic products such as some home appliances and industrial devices require professional installation services; these products include wall-mounted televisions, home theater systems, washers and dryers, refrigerators with water purifying systems, numerical control machines, network servers, and so on. Third, there is an organizational need of electronic vendors which makes the planning tasks of vehicle routing and scheduling for delivery and installation more complicated. The expense to maintain nationwide distribution and service network is too high to make it practical and economical. Therefore, electronic vendors adopt the practice of outsourcing the delivery to third parties and maintaining their own service departments or commissioning authorized service providers for the installation.

In the VRP under consideration, two types of demands in a complex electronics market are considered; one type requires only delivery of products, and the other requires both delivery and installation of them. It is not efficient to have the skilled professionals with installation capability to drive the delivery vehicles, haul the (potentially heavy) products, and provide the installation service. Therefore, electronics vendors prefer to operate separately two different types of vehicles (delivery and installation vehicles). It is assumed that delivery vehicles have identical loading capacity to carry the products and installation vehicles do not. Both types of vehicles start from a same single depot at the beginning and return to the depot within a specified time called *maximum operation time*. Delivery demands of all customers are known in advance. Based on the delivery demands, a set of customers is assigned to a delivery vehicle. The sum of customers' demands assigned to a delivery vehicle cannot exceed the loading capacity of the delivery vehicle. The delivery vehicle and (if needed) the installation vehicle are permitted to visit each customer only once. In addition, there is a constraint to satisfy the expected service quality, called *service level*, which is the maximum allowable time difference between the arrivals of delivery and installation vehicles. It is necessary for the installation vehicle to visit a customer within the predetermined maximum allowable time after the delivery vehicle's visit to that customer. Therefore, the synchronization of both types of vehicles needs to be carefully planned to guarantee the promised service level. Figure 1 shows a typical example of the VRP under consideration and its potential solution.

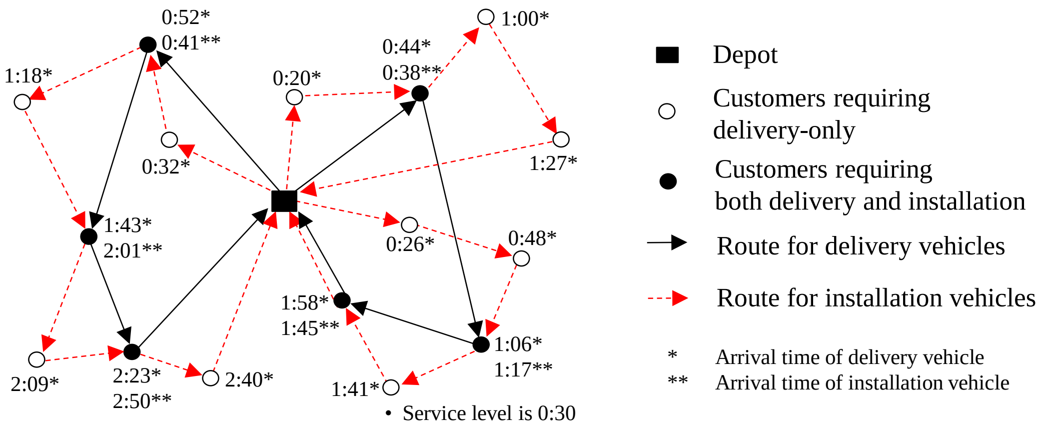


Figure 1. An example of the VRP under consideration

VRP is known as one of NP-hard problems in combinational optimization, it is difficult to obtain optimal solutions through general optimization methods owing to their high computational complexity (Bodin et al, 1983). The VRP in this paper the can be viewed as a combination of two traditional VRPs that have interactions. One is the VRP for delivery vehicles, showing the characteristics of capacitated vehicle routing problem (CVRP), and the other is the VRP for installation vehicles, showing the characteristics of vehicle routing problem with time windows (VRPTW). They are well-known variations and specializations of VRPs (Potvin and Rousseau, 1995; Laporte et al, 2000; Toth and Vigo, 2002; Pisinger and Ropke, 2007). The VRP in this paper is more complicated than other typical VRPs studied widely in the past, since there are two different VRPs which need to be timely synchronized to assure the quality of service for customers. To solve the VRP, Kim and Lee (2010) developed a mixed integer non-linear programming (MINP) model and a hierarchical approach using genetic algorithm (HGA). The MINP model provided the optimal solution but it was not practical for problem of large size in reasonable computational time. On the other hand, the HGA may not guarantee the optimal solution for problem but it could provide solutions for test problems of any size. Moreover, the HGA has the restricted solution space since its hierarchical structure. In this paper, an endosymbiotic evolutionary algorithm (EEA) is proposed to solve the VRP in a reasonable amount of time.

2 MINP MODEL FOR THE VRP

In this section, a mathematical programming model of the VRP under consideration is presented. The objective of this model is to minimize the shortest distance or time of travels by all delivery and installation vehicles. The decision variables provide optimal routes and schedules of delivery and installation vehicles. The VRP under consideration includes a depot (at location 0) and N customers (at locations 1 to N), among which customers belong to A require both the delivery and the installation ($|A| \leq N$). The notations for decision variables and parameters are explained in the following:

Decision Variables:

x_{ijp}	$\begin{cases} 1, & \text{if the delivery vehicle p travels from location i to location j} \\ 0, & \text{otherwise} \end{cases}$
y_{ijq}	$\begin{cases} 1, & \text{if the installation vehicle q travels from location i to location j} \\ 0, & \text{otherwise} \end{cases}$
e_i	Arrival time of the delivery vehicle at location i, $i \in A$
f_i	Arrival time of the installation vehicle at location i, $i \in A$
w_i	Waiting time of the installation vehicle at location i, $i \in A$
u_{ip}	Subtour prevention variables for x_{ijp}
v_{iq}	Subtour prevention variables for y_{ijq}

Parameters:

N	Number of customers
K	Number of delivery vehicles
S	Number of installation vehicles
A	Set of locations of customers requiring both delivery and installation
T_{ij}	Traveling time from location i to location j
FC	Fixed cost per vehicle
D_i	Demand of customer at location i
VC	Capacity of a delivery vehicle
R_i	Time to complete an installation at location i, $i \in A$
OT	Available operation time per shift for vehicles
SL	Maximum allowable time between delivery and installation (i.e., service level)

The problem under consideration is formulated as a mixed-integer nonlinear programming (MINP) model as follows:

Minimize Z

$$= \sum_{p=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} T_{ij} x_{ijp} + \sum_{q=1}^{S}\sum_{i=0}^{N}\sum_{j=0}^{N} T_{ij} y_{ijq} + \sum_{i=0}^{N} w_i + FC\left(\sum_{k=1}^{K}\sum_{j=1}^{N} x_{0jk} + \sum_{s=1}^{S}\sum_{j=1}^{N} y_{0js}\right)$$

Subject to

$$\sum_{p=1}^{K}\sum_{j=1}^{N} x_{ijp} \le K \quad \text{for } i = 0 \qquad (1)$$

$$\sum_{j=1}^{N} x_{ijp} \le 1 \quad \text{for } i = 0, \forall p \qquad (2)$$

$$\sum_{j=1}^{N} x_{ijp} - \sum_{j=1}^{N} x_{jip} = 0 \quad \text{for } i = 0, \forall p \qquad (3)$$

$$\sum_{p=1}^{K}\sum_{j=0}^{N} x_{ijp} = 1 \quad \text{for } i = 1 \ldots N \qquad (4)$$

$$\sum_{p=1}^{K}\sum_{i=0}^{N} x_{ijp} = 1 \quad \text{for } j = 1 \ldots N \qquad (5)$$

$$\sum_{j=0}^{N} x_{ijp} - \sum_{j=0}^{N} x_{jip} = 0 \quad \text{for } i = 1 \ldots N, \ \forall p \qquad (6)$$

$$\sum_{i=1}^{N} D_i \left(\sum_{j=0}^{N} x_{ijp}\right) \le VC \quad \text{for } \forall p \qquad (7)$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} T_{ij} x_{ijp} \le OT \quad \text{for } \forall p \qquad (8)$$

$$u_{ip} - u_{jp} + N x_{ijp} \le N\text{-}1 \quad \text{for } i \ne 0, j \ne 0, i \ne j, \forall p \qquad (9)$$

$$\sum_{i=1}^{N} x_{iip} = 0 \quad \text{for } \forall i, \forall p \qquad (10)$$

$$x_{ijp} = \{0,1\} \quad \text{for } \forall i, \forall j, \forall p \qquad (11)$$

$$\sum_{q=1}^{S}\sum_{j=1}^{N} y_{ijq} \le S \quad \text{for } i = 0, j \in A \qquad (12)$$

$$\sum_{j=1}^{N} y_{ijq} \le 1 \quad \text{for } i = 0, \forall q \qquad (13)$$

$$\sum_{j=1}^{N} y_{ijq} - \sum_{j=1}^{N} y_{jiq} = 0 \quad \text{for } i = 0, \forall q \qquad (14)$$

$$\sum_{q=1}^{S}\sum_{j=0}^{N} y_{ijq} = 1 \quad \text{for } i \in A \qquad (15)$$

$$\sum_{q=1}^{S}\sum_{i=0}^{N} y_{ijq} = 1 \quad \text{for } j \in A \qquad (16)$$

$$\sum_{j=0}^{N} y_{ijq} - \sum_{j=0}^{N} y_{jiq} = 0 \quad \text{for } i \in A, \ \forall q \qquad (17)$$

$$v_{iq} - v_{jq} + N y_{ijq} \le N\text{-}1 \quad \text{for } i \ne 0, j \ne 0, i \ne j, \forall q \qquad (18)$$

$$\sum_{i=1}^{N} y_{iiq} = 0 \quad \text{for } i \in A, \ \forall q \qquad (19)$$

$$y_{ijq} = \{0,1\} \quad \text{for } \forall i, \forall j, \forall q \qquad (20)$$

$$e_0 = 0 \qquad (21)$$

$$f_i \text{ -} \le SL \quad \text{for } i \in A \qquad (22)$$

$$w_i \le 0 \quad \text{for } i \in A \qquad (23)$$

$$\sum_{p=1}^{K} = 0 \quad \text{for } \forall j \qquad (24)$$

$$\sum_{q=1}^{S} \sum_{i} \cdots = 0 \qquad \text{for } j \in A \qquad (25)$$

$$\sum_{i=0}^{N} \sum_{j} \cdots = OT \qquad \text{for } \forall q \qquad (26)$$

The objective function of the given MINP consists of three parts. The first part of the objective function is the sum of the shortest traveling times of the vehicles, which is the major cost in the problem. The second part of the objective function is the sum of the waiting time of installation vehicles occurring due to the synchronization of two different types of vehicles. The last part is the fixed cost of vehicles in operation. For the last part, the fixed cost per vehicle (FC) can be considered as a certain penalty. The traveling distance or the transportation cost can also be used to optimize the model for the different purposes. The constraints can be classified into three different sets; the first concerning the VRP for delivery vehicles (constraints (1)~(11)); the second concerning the VRP for installation vehicles (constraints (12)~(20)); and the third concerning the synchronization for both types of vehicles (constraints (21)~(26)). Constraints (21)-(25) guarantee the quality of service by satisfying the service level (SL) for customers requiring both the delivery and the installation.

If an installation vehicle gets to a location earlier than a delivery vehicle, it sometimes needs to wait for the delivery vehicles before the installation starts. An installation vehicle may not leave immediately after the installation at a location to avoid waiting at the next location or to make other arrangements. The waiting time (w_i) of the installation vehicle at a location is defined as an amount of time spent before or after the installation at the corresponding location. We also tested another mathematical programming model by replacing constraint (23) with another constraint, $\max\{0, e_i - f_i\} - w_i = 0$, in order to remove the waiting time after the installation at a location. Since the third set of constraints includes nonlinear ones, the mathematical programming for the VRP under consideration is more complicated than the original VRP that can be formulated as a mixed integer programming (MIP) model.

3 ENDOSYMBIOTIC EVOLUTIONARY ALGORITHM FOR SYNCHRONIZATION OF DELIVERY AND INSTALLATION

A concept of endosymbiotic hypothesis to explain biological theories was proposed by Margulis (1980), and it is widely accepted in the area of biology. EEA can effectively solve complex problems in dynamic situations where multiple subproblems are interwoven (Kim et al., 2001; Kim et al., 2006). The original problem is split into subproblems, and each subproblem has a population consisting of a set of elements. An element in the population for a subproblem can be a part of a complete solution to the original problem. The VRP for delivery vehicles and the VRP for installation vehicles are regarded as subproblems in the problem under consideration. The EEA well manages two subproblems and efficiently acquires high quality solutions of the entire VRP. This paper provides a management strategy to efficiently plan the vehicle routes and schedules of vehicles for delivery and installation and to effective timely synchronize schedules of those vehicles. The proposed EEA generates and synchronizes the vehicle routing and scheduling plans for delivery and installation vehicles, while satisfying the constraints on the service level and the vehicle capacity, in order to minimize the sum of the traveling times of all vehicles.

The EEA is one type of symbiotic evolutionary algorithm (SEA) that can consider multiple subproblems at the same time (Potter, 1997; Moriarty and Miikkulainen, 1997; Kim et al., 2003). When the original problem is interwoven by multiple subproblems and we want to solve the original problem as a whole instead of considering them individually, the EEA can be a good option to search for the solutions of multiple subproblems concurrently. The EEA considers each subproblem along with its symbiotic partners, which are corresponding subproblems, in the original problem during the evolution. The concurrent consideration of subproblems may reduce the probability that the algorithm dwells on the local optima. The EEA tries to produce good solutions in balance among cooperative subproblems.

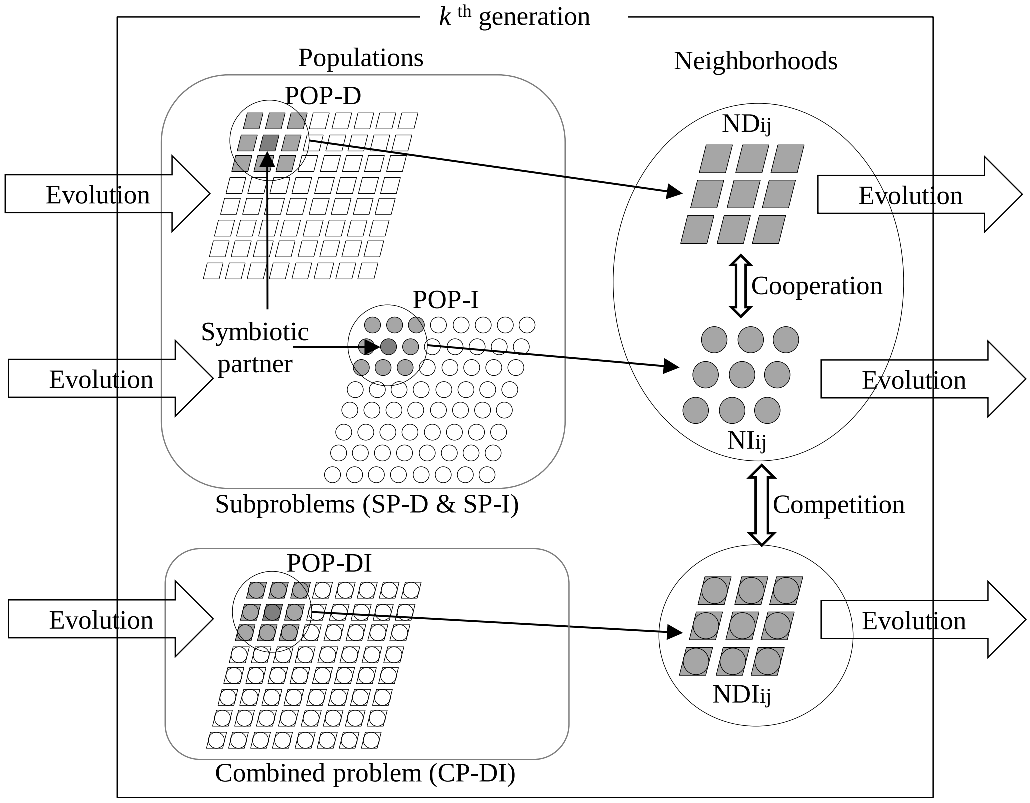


Figure 2. The concept of the EEA for the VRP under consideration

The EEA for the VRP under consideration breaks down the original problem into two different element subproblems and it also consider the original problem itself as combined problem of two subproblems; one is the subproblem for delivery vehicles (SP-D), another is the subproblem for installation vehicles (SP-I), and the last is the combined problem considering both delivery and installation vehicles (CP-DI). Two subproblems and the combined problems maintain their own population. The populations for the subproblems regarding delivery and installation vehicles in the proposed algorithm are referred as POP-D and POP-I, respectively. These two populations play the roles of corresponding symbionts in the endosymbiotic theory, and individuals in those populations represent partial solutions of the original problem. Both populations evolve in such a direction that corresponding symbionts from both populations cooperate with each other to find the better solutions to the original problem. Since individuals in POP-D and POP-I are merely partial solutions, only when corresponding symbionts are combined appropriately and feasibly into the endosymbionts does evaluation of the solutions to the original problem become possible. The population for the combined problem (POP-DI) plays the role of the endosymbiont in this algorithm. An endosymbiont carries the genes of all symbionts, which are partial solutions. Therefore, individuals in POP-DI represent solutions to the original problem. The individuals in POP-DI compete for survival with new offsprings that are generated through the mating of individuals from POP-D and POP-I. Eventually, POP-DI evolves toward a better population that contains better individuals in terms of the original problem. Figure 2 shows the concept of the EEA for the VRP under consideration.

In the algorithm, individuals in POP-D and POP-I are separately working as partial solutions but cooperate with each other. To make a complete solution of the original problem, a pair of individuals from each subproblem is required. On the other hand, individuals in POP-DI represent solutions of the original problem and they compete with those created by combination of individuals from POP-D and POP-I. In order to improve the search efficiency of the cooperative and competitive processes, the algorithm uses localized interactions among the populations instead of randomized and scattered interactions. For appropriate interactions for the processes among populations, topological locations of corresponding individuals are defined as follows.

Each population forms a two-dimensional structure of toroidal grid with the same number of individuals, and individuals in the population are mapped into the cells of the grid. An individual in the population has its own location index (x, y) and is surrounded by 8 neighbor individuals. The individual has corresponding individuals in the same geographical location at the toroidal grids of other two populations. Hence, when an arbitrary location (i, j) is selected at a generation, the neighborhoods of individuals including (i, j) at center of the 3×3 grid and 8 neighbor individuals in the POP-D, POP-I, and POP-DI are defined as ND_{ij}, NI_{ij}, and NDI_{ij}, respectively. Only the individuals in these three sets of neighborhoods are considered for the interactions among the three populations at the generation. The neighborhoods of ND_{ij} from POP-D and NI*ij* from POP-I cooperate to find a good solution of the problem. Since each neighborhood contains 9 individuals, 81 (9×9) combinations are considerable as the candidate solutions of the original problem. The best combination among them is compared with the current best solution in the algorithm that competes with 9 individuals in NDI_{ij} from POP-DI as well. Based on the interactions among the sets of neighborhoods, the parallel search with partial solutions from POP-D and POP-I in subproblems and the integrated search with complete solutions from POP-DI in the combined problem are carried out simultaneously in all generations. The procedures of the proposed algorithm proceeds as follows.

Step 1: Initialization of population
Generate initial individuals of POP-D, POP-I, and POP-DI, randomly. Set the best solution value,
$f_{best} = -\infty$.

Step 2: Construction of neighbors
Select an arbitrary location (i, j) and set up the neighborhoods, ND_{ij}, NI_{ij}, and NDI_{ij}, in each population.

Step 3: Cooperation between subproblems
Step 3.1: Evaluate the fitness of all possible combinations that can be produced by the concatenation of individuals in ND_{ij} and NI_{ij}.
Step 3.2: Let $d_p i_p$ be the best combination among those evaluated in Step 3.1, which becomes a candidate endosymbiont.
Step 3.3: If $f(d_p i_p) > f_{best}$, then update $f_{best} = f(d_p i_p)$ and keep $d_p i_p$ as the best solution from subproblems for the competition in Step 4.

Step 4: Competition between entire problem and subproblems
Step 4.1: Evaluate the fitness of individuals in NDI_{ij}.
Step 4.2: Let di_v be the best individual and di_w be the worst on in NDI_{ij}.
Step 4.3: If $f(di_v) > f_{best}$, then update $f_{best} = f(di_v)$ and keep di_v as the current best solution.
Step 4.4: If $f(d_p i_p) > f(di_w)$, then replace di_w with $d_p i_p$ in NDI_{ij}.

Step 5: Evolution
Step 5.1: Perform the evolution of individuals in ND_{ij}.

Step 5.2: Perform the evolution of individuals in NI_{ij}.
Step 5.3: Perform the evolution of individuals in NDI_{ij}.
Step 5.4: Release the evolved neighbors, ND_{ij}, NI_{ij}, and NDI_{ij}, to the population, POP-D, POP-I, and POP-DI, respectively.

Step 6: Iteration and termination
If the termination criteria of evolution are met, then stop. Otherwise, go back to Step 2 and repeat the process.

In Step 5, not only POP-DI but also POP-D and POP-I evolve to find better solutions through competition and cooperation in the algorithm. Only individuals in the neighborhood in three populations are considered to perform the evolution. Each population requires its unique genetic representation for the corresponding problem and goes through its own genetic operations to evolve. Genetic representations of individuals are very important because they are domain-specific and have large impacts on the performance of the genetic evolution. The details of genetic representations and operators are described in the following sections.

4 GENETIC REPRESENTATIONS AND OPERATORS FOR SUBPROBLEMS

In this paper, one-dimensional array is used to genetically represent individuals for SP-D and SP-I. According to the characteristic of each subproblem, the length of an individual is equal to the number of customers. For SP-D, let N be the number of customers for the VRP under consideration and n_i be the number of customers who will be assigned and served by delivery vehicle i. An individual consists of N genes (d_1, d_2, …, d_N). The genes contain the indices of customers requiring delivery. Since each customer is to be visited by only one delivery vehicle, the index of a customer must be shown only once in the genetic representation. The decoding procedure determines the routes and schedules of delivery vehicles in operation. In order to assign customers to delivery vehicles, the algorithm's greedy method considers the customers' demand, the loading capacity of delivery vehicles, and the maximum operation time. Figure 3 shows the individual before and after the decoding process. A set of customers, indicated by genes from $d_{v1,1}$ to $d_{v1,n1}$, is allocated to the first delivery vehicle (v_1). The remaining customers are assigned to other delivery vehicles in a similar manner. The genetic representation for SP-I is developed in a similar manner and omitted due to the redundancy. Another greedy method has been used to assign customers to installation vehicles. However, unlike the genetic representation of SP-D, the greedy method considers the arriving time of delivery vehicles for customers, the installation service time per customer, the service level, and the maximum operation time.

$$\lfloor d_1, d_2, d_3, \quad \cdots \quad , d_{N-2,}\ d_{N-1}, d_N \rfloor \Rightarrow \lfloor d_{v1,1}, \cdots, d_{v1,n1} \big| \ d_{v2,1}, \cdots, \ d_{v2,n2} \big| \quad \cdots \quad .$$

Figure 3. The decoding method of the subproblem for delivery

A pair of individuals from populations of subproblems, POP-D and POP-I, is required to make a complete solution for the VRP under consideration, because an individual from either subproblem is only a partial solution to the original problem. The propose algorithm has a single fitness function that requires a completely formed solution for the problem. Therefore, an individual solution from either subproblem is not complete to be evaluated separately. The evaluated function value of the pair of individuals from each subproblem is called *fitness value* which represents the survivability of them in the problem. The pair of individuals having a higher fitness value than others retains more chance to survive in the next generation. For the fitness value of a completely formed solution, which consists of individuals from both POP-D and POP-I, the traveling times of delivery vehicles and their fixed costs are first calculated for the individual from POP-D. While the traveling times of delivery vehicles are calculated, the arrival times of delivery vehicles at customers are also determined. Based on the arrival time of delivery vehicles at customers, the time windows for the arrival of installation vehicles and their fixed costs are calculated for the individual taken from POP-I. Then the traveling schedules of installation vehicles are determined, including the traveling times and waiting times of installation vehicles.

In order to calculate the fitness value of a pair of individuals from two subproblems, the followings are considered. Let τ_i be the sum of traveling times of delivery vehicles that are included in individual *i* of POP-D; $\varphi_{i,j}$ be the sum of traveling times of installation vehicles in individual *j* of POP-I (which are calculated based on the information on individual *i* of POP-D); $\omega_{i,j}$ be the sum of waiting times of installation vehicles in individual *j* of POP-I (again calculated based on individual *i* of POP-D); δ_i be the sum of fixed costs of delivery vehicles under operation in individual *i* of POP-D; and $\gamma_{i,j}$ be the sum of fixed costs of installation vehicles under operation in individual *j* of POP-I. ; Λ be the sum of unloading times by delivery vehicles at all customers; and I be the sum of installation times at customers requiring the installation. Since traveling times between any pair of customers are known, the traveling times of vehicles can be calculated from vehicles' visiting sequences. As mentioned previously, the waiting time occurs only when an installation vehicle arrives at a customer before a delivery vehicle arrives there, and it is calculated as the time lapse from the arrival of the installation vehicle to the completion of the unloading from the delivery vehicle. Fixed costs of delivery and installation vehicles are also computed by multiplying the known fixed cost per vehicle times the number of vehicles in operation. The sum of unloading times at customers (Λ) and the sum of installation times at customers (I) can be ignored since they are constants that can be omitted. Therefore, the fitness function for a combination of the individual *i* from POP-D and the individual *j* from POP-I , $\pi_{i,j}$, is defined as follows:

$$\pi_{i,j} = \frac{1}{\tau_i + \varphi_{i,j} + \omega_{i,j} + \delta_i + \gamma_{i,j} (+\Lambda + \mathrm{I})}$$

In each generation, the proposed algorithm searches higher quality solution of the VRP under consideration through the localized interaction among three neighborhoods, ND_{ij}, NI_{ij}, and NDI_{ij}. At the end of each generation, only the individuals in the three neighborhoods are considered for genetic operations such as selection, crossover, and mutation to reproduce new individuals for the next generation. Since a pair of individuals from each subproblem is needed to make complete solution for the original problem, the individual in the pair share the fitness value in the proposed EEA. An individual which has a higher fitness value than others has a higher chance to participate in reproduction. The probability of an individual to be selected for reproduction is calculated by the fitness value of the individual over the sum of all individuals' fitness values in the population.

As reproduction processes, crossover and mutation procedures which are used in the proposed algorithm, are described in the following. Due to the simple, fast greedy decoding procedure, a premature convergence has been observed, where individuals have different genetic representations but the same assignment of customers to vehicles. Hence, a hybrid crossover procedure has been used to efficiently and effectively reproduce new offspring from two parents in the current population. The proposed crossover procedure proposed to consider both the inheritance from parents and the diversity of the population, while avoiding premature convergence of the population. The procedure of the hybrid crossover is described as follows.

Step 1: Two individuals (P1 and P2) in the current population are randomly selected.
Step 2: The corresponding genes, which are a series of location indices for the chosen

vehicle, are copied into an offspring in the same order as they appear in P1, and the corresponding genes are deleted from P1.

Step 3: The remaining genes in P1 are rearranged in order.

Step 4: A gene is randomly selected from the remaining genes in P1.

Step 5: The genes in the pre-cut section of P1 are added into the offspring in the same order in which they appear in P1.

Step 6: The indices of already-inherited genes from P1 are deleted in the other parent (P2).

Step 7: The remaining genes in P2 are copied into the offspring in the order in which they appear in P2.

The ranking replacement strategy has been used to construct the population for the next generation in the proposed algorithm (Chu and Beasley, 1998). In the course of crossover procedures, newly generated offspring competes with all individuals in the current neighborhood. That is, let π_{off} and π_w be the fitness values of the new offspring and the worst individual in the current neighborhood, respectively. If $\pi_{off} > \pi_w$, the corresponding offspring will replace the worst individual in the current population.

An exchange mutation procedure has been used in the proposed algorithm by swapping two genes from an individual selected randomly at the mutation rate (Gen and Cheng, 2000).

5 GENETIC REPRESENTATIONS AND OPERATORS FOR THE COMBINED PROBLEM

The proposed EEA consists of not only subproblems (SP-D and SP-I) but also a combined problem (CP-DI). An individual in POP-DI is a candidate solution for the original problem and consists of two different parts of genomes; one has an identical form of the individual in POP-D, and the other has an identical form of the individual in POP-I. Therefore, an individual in POP-DI is a completely formed solution for the VRP under consideration. The genomes for the delivery and the installation are concatenated in a row. The genome for the delivery is placed at the head of the individual and is followed by the genome for the installation. The genome at the head and the genome at the tail use identical genetic representations for SP-D and SP-I, which are already explained in Section 4. The concatenation of two different genes results in a one-dimensional array, but two separate genes within an individual in POP-DI are heterogeneous. Therefore, the genetic operations for POP-DI are carried out separately in two steps: one for the head and the other for the tail. The hybrid order crossover operation, which has been explained earlier, is executed over the head and then over the tail, separately. The exchange mutation operation also needs to be performed in two steps, in a similar way. A detailed description of this operation is omitted to avoid redundancy.

The fitness of an individual in POP-DI is evaluated sequentially over the head and the tail. The genome at the head is decoded to generate schedules and routes for delivery vehicles. Based on the schedules for delivery vehicles, the genome at the tail is decoded to generate schedules and routes for installation vehicles. Finally, the integrated schedules and routes for both types of vehicles become the complete solution of the individual in POP-DI. From the schedules and the routes of delivery and installation vehicles, the sum of traveling times of delivery vehicles, the sum of traveling times of installation vehicles, and the sum of waiting times of installation vehicles are calculated. Furthermore, from the results of the decoding procedure, the number of vehicles to be used is also determined.

The evaluation of the fitness values for individuals in POP-DI is conducted in a similar manner to those for the subproblems. For the individual k in POP-DI, let γ_k, be the sum of traveling times of delivery vehicles, φk be the sum of traveling times of installation vehicles,

ω_k be the sum of waiting times of installation vehicles, δ_k be the sum of fixed costs of delivery vehicles, γ_k be the sum of fixed costs of installation vehicles, Λ be the sum of unloading times by delivery vehicles for all customers, and I be the sum of installation times at customers requiring installation. Since traveling times between any pairs of customers are known, the traveling times of vehicles can be calculated from vehicles' visiting sequences. As already noted, waiting time occurs only when an installation vehicle arrives at a customer before a delivery vehicle arrives there, and it is calculated as the time lapse from the arrival of the installation vehicle to the completion of the unloading from the delivery vehicle. Fixed costs of delivery and installation vehicles are also computed by multiplying the known fixed cost per vehicle times the number of vehicles in operation. The sum of unloading times at all customers (Λ) and the sum of installation times at customers requiring the installation (I) are constant and can be omitted. From the definitions, the fitness of the individual k in POP-DI (π_k) is defined as follows.

$$\pi_k = \frac{1}{\tau_k + \varphi_k + \omega_k + \delta_k + \gamma_k (+\Lambda + \mathrm{I})}$$

The fitness functions for both the subproblems and the combined problem are exactly same and they are is designed to reduce not only the sum of traveling times of all vehicles but also the sum of waiting times of the installation vehicles while evolving.

6 COMPUTATIONAL RESULTS AND CONCLUSION

In order to show the performance of the proposed EEA, the search capability of the proposed EEA to a test problem has been compared and analyzed with another approach, HGA. In HGA for the VRP under consideration, a VRP for delivery vehicles (VRP-D) is solved first, which is followed by solving a VRP for installation vehicles (VRP-I). The GA for VRP-D obtains a solution for delivery vehicles which is used as input to solve VRP-I and then the GA for VRP-I obtains a solution for installation vehicles based on the input from VRP-D (Kim et al, 2012). The solutions from both sub VRPs become the solution of HGA for the VRP under consideration.

In the test problem, customers are randomly located in a 100×100 square area and a single depot is located in the center. The test problem consists of 30 customers requiring the delivery while 10 customers require the installation. The delivery demand of each customer is between 2 and 10; there are 8 delivery vehicles at the depot and the loading capacity of delivery vehicles is 20; there are 3 installation vehicles at the depot; the installation time and the service level are 10 and 60, respectively; and the maximum operation time is 480.

All computational experiments are carried out on a personal computer with a 3.4 GHz Pentium 4 CPU and 2.0 GB RAM. The algorithms for the HGA and the EEA are implemented in Visual Basic programming language using the Microsoft Visual Studio.NET Framework 1.1 version.

For the genetic evolution in the proposed algorithm, the following parameters are used. A 10×10 toroidal grids is used for POP-D, POP-I, and POP-DI. The crossover rate and the mutation rate are set to 0.8 and 0.05, respectively. When the current number of generation reaches at a specified maximum number depending on the size of solution space, the proposed algorithm terminates.

Figure 4 shows the best solutions produced by two approaches to the test problem. Each result uses 8 delivery vehicles and 2 installation vehicles to provide services. The sum of the travelling times of all vehicles from the HGA and the EEA are 1111.43 and 1077.43, respectively. Since, the routes of installation vehicles in the HGA depends on the schedules of delivery vehicles, their routes may be twisted. However, since the proposed EEA consider routes and schedules for both types of vehicles at the same time, it has less chance to get twisted routes of any type of vehicle. The routes from and to the depot have been omitted for the simplicity in Figure 4.

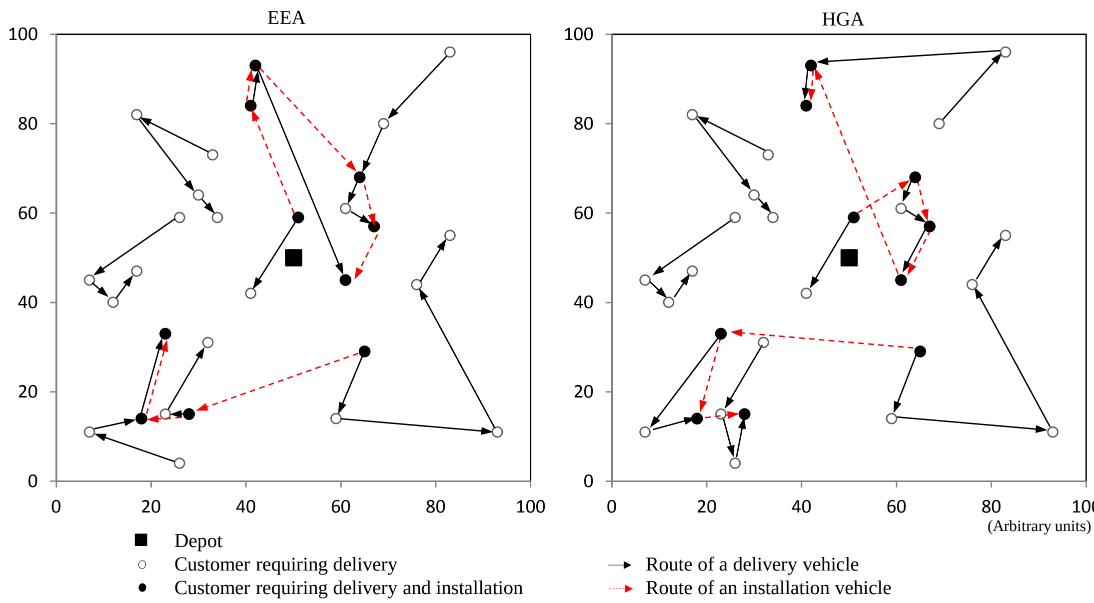


Figure 4. The experiment results from the HGA and the EEA

In addition, the progresses of the best solutions in both algorithms have been compared. A large test problem which consists of 100 customers requiring the delivery while 50 customers requiring the installation is used for the experiment. Figure 5 shows the progress of the best solutions in each generation for the test problem from the EEA and the HGA. Lines in the figure represent the change in the value of the best solution in both approaches during the evolution. Until the 2,500th generation, the HGA does not have any solution of the test problem since the algorithm only considers the VRP-D. As shown in Figure 4 and 5, the proposed EEA shows better performance than the HGA to the test problems.

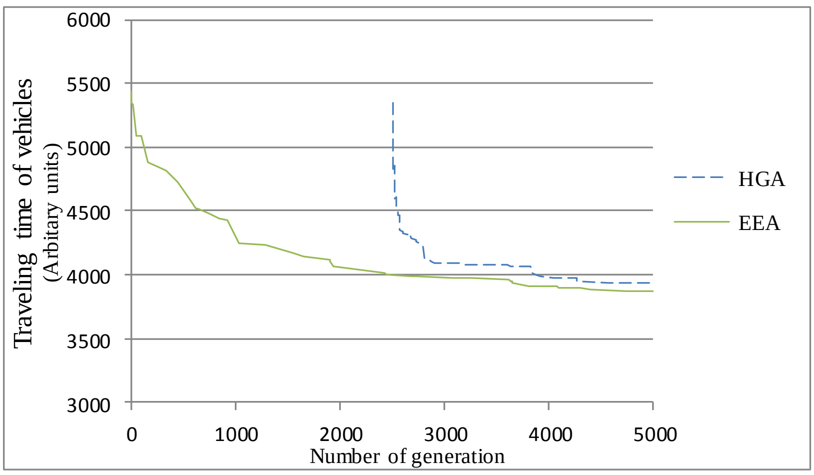


Figure 5. Progresses of the EEA and the HGA

In this paper, a VRP found in the electronics market industry has been identified and introduced. Two types of customers' demands for delivery and installation are considered in the problem. One type requires only the delivery, and the other requires both the delivery and the installation. In order to satisfy both demands, electronics vendors prefer to operate separately two different types of vehicles (delivery and installation vehicles). A service quality requirement, measured by the time lapse between delivery and installation at a customer location, must be provided. The installation vehicle must visit a customer within the predetermined maximum allowable time after the delivery vehicle's visit to that customer. Therefore, the synchronization of both types of vehicles is required to ensure the guaranteed quality of service for customers ordering both the delivery and the installation. Installation vehicles can visit customers earlier than delivery vehicles, resulting in waiting times for installation vehicles at the corresponding customer locations.

The EEA is proposed to efficiently solve the problem under consideration. Appropriate genetic representations, fitness functions, genetic operators, generic parameters for subproblems and the combined problem in the algorithm have been developed considering the characteristics of the problem. Furthermore, the construct method of populations for subproblems and the combined problem and the interaction among them have been introduced. The computational results show that the proposed EEA is able to find the optimal or near-optimal solutions to problems of various sizes. The results of the algorithm to test problems are compared with other approaches.

In future research, various industries with different characteristics can be studied to model theirs unique VRP. Unlike the problem under consideration in this paper, delivery and installation vehicles can be operated as a single business function. In other words, a vehicle can provide both delivery and installation. It may be interesting to compare those two cases in order to justify the economical merits.

REFERENCES

Bodin L D, Golden B L, Assad A A and Ball M (1983). Routing and scheduling of vehicles and crews, the state of the art. *Comput Opns Res* **10:** 63-212.

Chien S and Spasovic L (2002). Optimization of grid bus transit systems with elastic demand. *J Adv Transport* 36: 63-91.

Chu P C and Beasley J E (1998). Constraint handling in genetic algorithms: the set partitioning problem. *J Heuristics* **4:** 323-357.

Gen M and Cheng R (2000). *Genetic algorithms and engineering optimization*. John Wiley & Sons, Inc.

Golden B L and Wasil E A (1987). Computerized Vehicle Routing in the Soft Drink Industry. *Opns Res* **35:** 6-17.

Ji P (2007). The vehicle routing problem: The case of the Hong Kong postal service. *Transport Plan Tech* **30:** 167-182.

Kim C K and Lee S W (2010). Hierarchical approach to Vehicle Routing Problem for Delivery and Installation, *Proceeding of The 2010 International Conference on Logistics and Maritime System, LOGMS* 2010, Busan, Korea.

Kim C K, Ji U S, and Lee S W (2012). Hierarchical approach to Vehicle Routing Problem for Delivery and Installation, *International Journal of Industrial Engineering: Theory, Applications and Practice* (in press)

Kim J Y, Kim Y and Kim Y K (2001). An endosymbiotic evolutionary algorithm for optimization. *Applied Intelligence*, 15, 117-130.

Kim Y K, Kim J Y, and Kim Y (2006). An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines. *Eur J Opl Res*, 168, 838-852.

Kim Y K, Park K, and Ko J (2003). A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Comput Opns Res*, 30(8), 1151-1171.

Laporte, G., Gendreau, M., Potvin, J-Y., and Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, *7*, 285-300.

List G and Mirchandani P (1991). An Integrated Network/Planner Multiobjective Model for Routing and Scheduling for Hazardous Materials and Wastes. *Transport Sci* **25:** 146-156.

Margulis L (1981). *Symbiosis in cell evolution*. San Francisco, CA: W. H. Freeman

Moriarty D E and Miikkulainen R (1997). Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, 5, 373-399.

Pisinger D and Ropke S (2007). A general heuristic for vehicle routing problems. *Comput Opns Res* **34:** 2403-2435.

Potvin J Y and Rousseau J M (1995). Exchange heuristic for routeing problems with time windows. *J Opl Res Soc* **46:** 1433-1446

Potter M A (1997). The design and analysis of a computational model of cooperative coevolution. (Unpublished doctoral dissertation). George Mason University, Arlington, VA.

Prive J, Renaud J and Boctor F (2006). Solving a vehicle-routing problem arising in soft-drink distribution. *J Opl Res Soc* **57:** 1045-1052.

Ripplinger D (2005). Rural school vehicle routing problem *Transport Res Rec* **1922:** 105-110.

Toth P and Vigo D (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Appl Math* **123:** 487-512.

Zografos K G and Androutsopoulos K N (2004). A heuristic algorithm for solving hazardous materials distribution problems. *Eur J Opl Res* **152:** 507-519.

MULTI AGENT SIMULATION OF A WAREHOUSE

Christoph Schwarz[1], Axel Hahn[2] and Jürgen Sauer[2]

[1]*OFFIS – Institute for Information Technology, Escherweg 2, 26121 Oldenburg, christoph.schwarz@offis.de*

[2] *Carl von Ossietzky University Oldenburg, Ammerländer Heerstr. 114-118, 26129 Oldenburg, axel.hahn@uni-oldenburg.de, juergen.sauer@uni-oldenburg.de*

ABSTRACT

Multi-agent-based simulations are increasingly studied in recent years. Such simulations are generally adaptable and easily parallelizable. A major advantage of these systems is the possibility to model individuals, for example individual drivers of reach stacker trucks who make different decisions about which order to pick next. Material handling systems for container ports and related logistic sceanrios are nowadays largely simulated by discrete event simulations. In this paper, we show that a simulation of these systems as a multi-agent based simulation with individual decision makers is possible. With this simulation the influence of individual decisions on the system can be studied. All decision-makers (ports, cranes, etc.) were simulated by agents. Beside presenting our approach we also present an evaluation with an existing stockyard.

Keywords: Simulation, Multi Agent Systems, Warehouse, Conveyors

1 INTRODUCTION

Material Flow Systems (MFS) are today usually studied with the help of discrete event simulation (DES) (see [Wenzel 2006] or [Kuhn and Wenzel 2007]). A variety of such tools is on the market today (see [Wenzel and Noche 2000]) which demonstrates their importance again. Plant Simulation[1] and AutoMod[2] are two examples and are widely used.

In discrete event simulations, the events are usually represented as flows of goods at fixed times. With these simulations bottlenecks within the material flow system and feasible clock rates for the facilities may be identified. In DES the system analyst models the concrete entities which are involved in the logistics resp. material flow such as cranes, reach stacker, conveyor belts, hubs, picking stations, floor conveyors, automated guided vehicles (AGVs), etc. Although these entities may be of different types, they do not have a genuine individuality. (See for example [Bangsow 2008])

Another method for the simulation of processes in intralogistics and material flow systems is the use of Petri nets (see [Reisig and Rozenberg 1998] and [Li *et al* 2006]). Again, there are programs to implement simulations with Petri nets such as Renew (cf. [Kummer *et al* 2004]).

In recent years multi-agent based simulations (MABS) are increasingly in the focus of research. Davidsson concludes reasons for this:

> *"However, compared to other approaches, e.g., traditional discrete event simulation, object-oriented simulation and dynamic micro simulation, MABS has a number of interesting properties which makes it useful also for other domains. For instance, it*

[1] www.plant-simulation.com
[2] www.automod.de

supports structure preserving modeling of the simulated reality, simulation of pro-active behavior, parallel computations, and very dynamic simulation scenarios. It is argued that MABS is a useful technique for simulating scenarios also in more technical domains. In particular, this hold for the simulation of technical systems that are distributed and involve complex interaction between humans and machines." (from [Davidsson 2001])

Several publications have provided evidence that for intralogistic scenarios MABS are slower than DES but, on the one hand, can be adapted more easily and, on the other hand they are easily parallelizable (see [Becker *et al* 2006] and [Gehrke *et al* 2006]). Since the simulation speed plays no significant role for our question, the advantages of MABS outweigh compared to DES systems.

In the area of MFS, some results were published on the basis of MABS. For example, Follert Roidl studied with the help of multi-agent systems, whether large-scale handling systems can be controlled by distributed systems (see [Follert and Roidl 2008]). The idea followed by Günthner and ten Hompel called "Internet of things in intralogistics" (see for example [Günther and ten Hompel 2010]) uses multi-agent systems in many studies as Scholz-Reiter does in investigating the possibilities of self-organization in logistic processes (see [Scholz-Reiter and Höhns 2006]).

Our question is, however, the impact of individual decisions made by individual drivers and the individual job allocation decisions rather than the material flow as a whole. More precisely, we want to examine the effects of various alternative strategies and cooperation of conveyors in a warehouse or a container port, paving the way for a cognitive logistics system ([Beth *et al* 2011] and [Ommen *et al* 2010). In the simulation, in addition to these strategies, the topology of the warehouse or the port, the various types of conveyors and various load profiles play a role.

To investigate these aspects as a first step, we have developed a MABS, in which it is possible that individual truck drivers can make individual decisions about which order to pick next or about which route to take. We also build an instance off this simulation with an existing warehouse. In this paper we present this simulation and the results of the simulation of this warehouse.

1.1 Paper Organization

In section 2 our requirements for the simulation system are presented. The first part of section 3 gives an introduction to the agent-based simulation library MASON, which we use as a basis for our approach. Next, the structure of our simulation is shown. In section 4 a comparison between simulation and reality is presented. In section 5 a conclusion is given and possibilities for following studies are described.

2 REQUIREMENTS FOR THE SIMULATION SYSTEM

For our research questions, it is necessary that the simulation can model individuals. These individuals should be able to answer the following questions independently: what jobs I take on, or about what jobs I am applying? What route do I take? Exactly these points cannot be achieved with DES systems, as they are on the market today and are therefore the most important requirement for the simulation system. It should be easily possible to model these behaviours separate for each agent and to easily change them.

Besides the ability of modelling individuals the simulation systems should fulfil the requirements on a simulation system for warehouse logistics in general. Such requirements include an easy modelling of the scenarios and the load profiles, automatic calculation of indicators and an visualization of the simulation.

3 SIMULATION

To fulfil all the requirements we had for the simulation system we build up a MASON based simulation system. This chapter describes the structure of this system. Section 3.1 presents a brief overview of the used simulation library MASON. In section 3.2 the modelling of the scenario is described.

The behaviour of the individual cranes and the disposition of goods can be achieved by a simple change in the respective corresponding agents.

3.1 MASON

To simulate the movement of containers on the yard or goods in the warehouse, the java library MASON (see [Luke *et al* 2005]) was selected, which was developed specifically for the construction of multi-agent simulations.

The declared aim of MASON is the simplification of simulations of complex adaptive systems, evolutionary algorithms, methods of self-organization, the physical modelling and machine learning. The homepage of MASON[3] contains a wide variety of projects in these areas, which are using the library.

The MASON library is modular. The modules can be seen in figure 1. The core of the model library provides the basic tools of MASON. These are essentially a scheduler for temporal control of agents and so-called grids which allow the modelling of the environment in which the agents move. One advantage of MASON compared to other simulation frameworks is the strict separation between the model and visualization. The visualization is provided by the MASON GUI tools. These tools enable a dynamic addition, modification and removal of visualization objects to agents and elements of the environment.

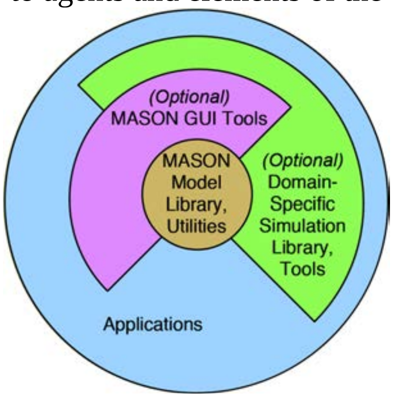


Figure 1. MASON modules (source: [3])

The visualization is used in our case mainly to illustrate the simulation of scenarios and for troubleshooting. Figure 2 shows a section of a screenshot from the simulation.

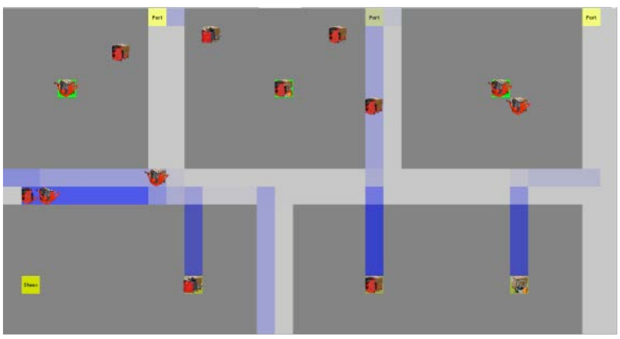


Figure 2. Simulation of a yard (section)

[3] www.cs.gmu.edu/eclab~/projects/mason

3.1.1 Scheduler

The scheduler implements a step-based simulation.

Methods of an agent can either be run once in a given simulation step in the future or in a sequence at specific time intervals. To implement an agent that is to be controlled by the MASON scheduler, it must implement the method step from the interface Steppable. Then this class can be marked for execution. The agents then have to calculate their actions in their step method. In principle, more is not needed to build an agent in MASON.

The scheduler is used in our simulation environment for all of the active agent classes. These are the conveyors (carrier, see section 3.2.4) and the ports (see section 3.2.1). The timing of the load generation (see section 3.2.3) is also performed by the scheduler. Furthermore, the scheduler is used for the control of the disposition of goods (see Section 3.2.5).

3.1.2 Grids

Grids are used for modelling the environments in MASON. They organize fields in 2- or 3-dimensional matrices. A field contains attributes (e.g. an integer or double value) or objects.

Grids include methods for the determination of proximity relationships. Thus, for example, objects can be determined which are in a specified radius around a particular field. These features are very useful to find other agents to be cooperating with or to communicate with. A simulation model can consist of several stacked grids. Neighbourly relations cannot be determined across different grids.

In our simulation environment, several grids of different types are used. The most important grids are an ObjectGrid2D to model the static aspects of the yard as roadways, ports and storage areas, a SparseGrid2D for modelling the dynamic aspects of the warehouse as the vehicles and two DoubleGrid2Ds to display scheduled and used track assignments and for the presentation and evaluation of vehicle conflicts (collisions).

3.2 Modelling the Scenarios

The modelling of our scenario consists of a model for the environment and of models for the moving objects in the environment. Our environment consists primarily of ports and transport routes.

Figure 2 displays an example of the visualization of our environment. It shows a yard with transport routes (light grey) and ports (yellow and green).

Ports are goods transfer points. They are responsible for the identification of goods, for the communication with host systems (e.g. TOS or in other cases EPR systems) and for the distribution of orders to transport vehicles. Container or Goods can only be transferred from the outside into the hall or vice versa at these points. In addition, ports serve as buffer areas within the hall.

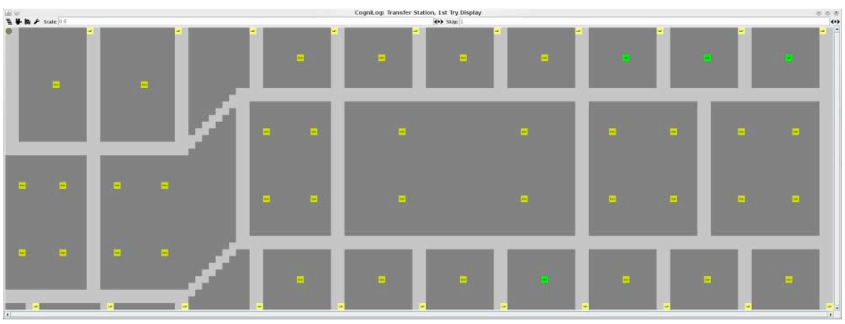


Figure 3. Simulation of the environment

3.2.1 Ports and Transport Routes

In the simulation the floor plan of a yard is modelled by a grid with equal-sized square elements. For each grid field an object accurately represents the particular area in the hall such as a port or a part of a route. The scale of the grid can be chosen. In the example in Figure 3, the fields have an edge length of 1 m. In addition, the grid has a graph as an overlay that describes the intersections and the travel routes (see Section 3.2.2). The objects that are annotated to the fields represent the conditions of the yard at this field.

Table 1. Objects in the grid

Object type	Java object	Active?
Inport	InPort	active
Buffer	BufferPort	active
Outport	OutPort	active (for indicators)
Driving area	DrivingArea	passive
Non driving area	NonDrivingArea	passive

Possible objects are listed in Table 1. Three types of objects, that are three port types, are active. This means that they show a behaviour controlled by the MASON scheduler. The behaviour of the OutPorts is trivial and merely serves to identify indicators. InPorts and BufferPorts have a more complex behaviour. Both types are responsible for the management of the goods which are placed on them. This includes the generation of transport orders and the disposition of these orders to the vehicles. The two object types DrivingArea and NonDrivingArea are passive, they do not have an own behaviour. Both, active and passive objects, are visualized.

PortFloor is the common abstract superclass of all ports. Instances of derived classes define methods to transfer goods (cargo) to ports (takeCargo), to start an auction process (transportProposalStart) and to end the current auction (transportProposalEnd). FloorOutPort requires no auctions, since the OutputPort do not buffer goods. It only receives goods, calculates indicators and removes the goods from the simulation environment.

Goods enter the warehouse always at an input port. In the simulation this is achieved with the Delivery of the InPort (which represents for example a truck). Each InPort regularly controls in its step method whether more capacity for goods is available at the port. If this is the case new goods are taken out of the delivery and placed at the CargoContainer of the port. Depending on the port type (FloorInPort or FloorBufferPort) the port will immediately generate a transport order, or will store the goods for later transportation.

3.2.2 Graphs

The graph describes the ways on which the vehicles can drive on the yard. Furthermore the graph contains information about the current and the planed occupancy of road segments. The route map is represented as a directed graph $G = (V, E)$, where each intersection, each branch and each endpoint of a way is a node in V. The directed edges of the edge set $E \in V \times V$ connect nodes and form the ways in the warehouse.

Static attributes are annotated on both the nodes from V and the edges from E. One attribute of a node is a $p \in \mathbb{R} \times \mathbb{R}$, which describes the position in space. Edges have the attributes $distance$, V_{max} and M_{max}, which describe the length, the maximum speed and the maximum load capacity of the route.

In addition, a set of assignments O is annotated on the edges as a dynamic attribute. An assignment $o \in O$ is a tuple $o = (carrier, t_{start}, t_{end}, p)$, that describes which vehicle ($carrier$) is planning to enter the edge at the time t_{start} and leave it at the time t_{end}.

Furthermore p gives the likelihood that the vehicle will actually perform the driving on this route at the given time.

3.2.3 Load Generation

The boundary of the system under consideration is located at the input and output ports. At the input port, goods are identified and prepared for transport, at the output port they are loaded onto trucks/trains etc.. Since handling processes of the yard are neglected, goods must be generated within the ports in a realistic manner. That is the task of load generation. Two load generators were implemented. One database related and one profile related.

The first generator is based on a database which includes information about the movements of goods through the warehouse. In each time step of the simulation, the generator checks with the help of this database whether new goods were taken into the hall at the current time step. If that is the case, these containers / goods are then generated in the input ports.

The database describes not only time relevant data like entering and leaving time but also information about characteristics of the goods themselves, such as weight and volume. These data are annotated to the goods in the simulation.

The second generator is based on a stochastic profile. This profile has probabilities for different cargo types to be generated at given time steps. These probabilities can be related to different times (e.g. weekdays vs. weekends, morning vs. noon, vs. afternoon vs. night). At each time step the different cargos are generated according to their probabilities.

3.2.4 Transport Vehicles

In the simulation, also cranes are responsible for the transportation of goods. In a warehouse conveyors are an equivalent. These cranes or conveyors always have a cargo space on which the goods are stored during the transport. This has been modelled by a cargo area which encapsulates all the essential properties of this area (geometrical dimensions, maximum weight, etc.). The cranes or conveyors also have a graph that describes the transport routes in the area, a plan that describes the following operations to be performed as well as other attributes that describe the position and the physical dimensions of the crane or conveyor. Even basic methods for the planning and execution of transports are defined here. Conveyor is an abstract superclass for cranes or conveyors. This decision should make it easier to simulate also continuous conveyors in later scenarios.

Discontinuous conveyors extend conveyors by adding attributes and methods for description and manipulation of physical parameters (e.g. the speed), for acquisition of transport orders and for the creation and execution of transport plans.

Conveyors are modelled as agents whose goal it is to acquire transfer orders and to execute them. To fulfil these goals, they need information from their environments. These information are used in conjunction with the internal state of the agent in order to plan and execute tasks. Ultimately, the planned actions are translated into movements which are executed by the conveyors. These movements then alter the environment.

The job allocation takes place in cooperation with the input ports and the buffer ports which hold open orders. Depending on the chosen strategy, these orders can be directly assigned to the trucks or can be placed in an auction process. Section 3.2.5 describes the auction process in more detail.

At the beginning of the execution of an order, the vehicles plan the route from their current location to the location of the goods and from there to the destination. To calculate the route the vehicles take the current and future truck route assignments into account, which it gets as input from its environment. It also can add individual preferences into account. Examples for such preferences could be as less turns or as less as possible other vehicles in the area. In other words each agent can have is own method of calculating its routes.

The vehicle than allocates the way segments that lie on the so-determined optimal route. Afterwards, the occupancy information of the environment gets updated. Since the occupancy information of the environment, which are calculated by adding the assignments of all vehicles, is dynamically changing in this way, the vehicles regularly needs to perform replanning during the execution. This behaviour corresponds to the observation of the situation in the warehouse by human drivers.

The execution determines necessary movement and manipulation of goods operations (picking up and setting down of goods) and puts them in order. Plan changes caused by new plans are also considered here.

3.2.5 Order Disposition

An auction based order disposition was developed. In this auction process open transfer orders are distributed through interaction between ports and vehicles. The initiative emanates from the ports. An auction is started when a good is due to be delivered to an output port. The port starts the auction by offering a new proposal which has a specific duration. The proposal, which is also modelled as an agent, calculates the time at which the auction will end and enters his own step method in the MASON scheduler to be informed at that time. In addition, all vehicles will be informed about the new transport offer.

Now it is up to the vehicles to bid on the proposal. They only do that when they are currently not involved in any other transport order and when they are capable of carrying out this transport in terms of weight and size of the goods. Finally a vehicle can decide not to bid on a proposal due to individual preferences (e.g. do not bid on proposals in the eastern part of the warehouse or do not bid on proposals that are far away from my current position).
In the case a vehicle wants to bid on a proposal it calculates the time it would need to execute the order (using the information from the environment and its individual routing algorithm) and sends this time to the proposal.

The end of an auction is initiated by calling the step method of the proposal. Then the method ProposalEnd is called in the port, where the bids are evaluated. The port choses the best bid, namely that one with the lowest costs. Afterwards, it sends all vehicles which had participated in the auction a message: the winner will receive a ProposalAccept message and the others receive a ProposalReject message.

4 EVALUATION

To evaluate our simulation system we build a simulation of a yard and a warehouse. For the later we have used the log of a warehouse of one of our industry partners. Due to the completeness of case the results for the warehouse are discussed here. The design of the warehouse was available and the movements of all goods from 2009-07-01 until 2010-07-01 were collected. The hall has four imports, four outports, and 36 buffer ports. In total 1,388,761 packages were moved during the given interval in the warehouse. For these packages, the following information was provided to us in a database: loading and unloading time stamps, volume, weight, inport and outport. So we used the database related cargo generator for this simulation (see. 3.2.3). Also the rules for order picking and route calculation in the given warehouse were known. We used these rules to model the behaviour of the agents (see. 3.2.4).

From the given database important logistic indicators such as delivery reliability, lead time and utilization ratio of the conveyors could easily be determined. We had to select indicators in order to obtain a comparison between the data made available by the industry partner and the data of the simulation. The indicators listed in Table 2 were used by us for that comparison.

The aim was to replicate the processes in the warehouse in our multi-agent based simulation. Table 3 shows that the simulation reflects the real situation to a large extend. When looking at the indicators it is noticeable that some values differ slightly, including the delivered packages, which differ by 1,229. This difference can be explained by the fact that packages in the database are removed before the end of simulation, and therefore could not be accounted for in the simulation. Similarly, the average processing time per package in the simulation is slightly increased: namely 55 delays cause an average delay of about 2 minutes per package.

Subtracting the average delays from the processing time, a lead time of 11:56:38 is reached, which only causes a difference of 39 seconds. Analogously, the slight difference in the average storage time for each package can be explained.

Table 2. Indicators of the comparison

delivered packages	Integer
delayed packages (absolute)	Integer
delayed packages (relative)	$\frac{delayed\ packages}{delivered\ packages}$
average delay per package	$\frac{\frac{\sum delay}{delayed\ packages}}{\sum ports\ with\ delay}$
average transportation time per package	$\frac{\sum transportation\ time}{delivered\ packages}$
average storage time per package	$\frac{\sum storage\ time}{delivered\ packages}$

Table 3. Comparison between simulation and reality

Indicator	Reality	Simulation
delivered packages (absolute)	1.388.761	1.387.532
delayed packages (absolute) delayed packages (relative)	0 0%	55 0,004%
average delay per package (sec)	0	130
average transportation time per package (hh:mm:ss)	11:55:59	11:58:48
Average storage time per package	11:59:46	11:55:48

4 CONCLUSION

The presented multi-agent-based simulation is capable of simulating the operations of a yard or a warehouse. All relevant indicators have been achieved within the tolerance. Furthermore, all the decision makers were simulated as agents, which are capable of individual order picking decisions and of individual routing algorithms. Therefore it is possible to investigate the impact of individual decisions in a warehouse with the help of the presented simulation system.

In the future we will investigate the impact of such individual decisions on the important logistic key indicators. Some questions that arise here are: is it useful that the vehicles have different order picking behaviours (for example that some vehicles only pick orders nearby and some only orders which are urgent)? Can it be useful that some vehicles avoid the middle of a warehouse?

Furthermore different conflict resolution options will be implemented in the simulation system. Conflicts can occur in order for route planning and scheduling (avoid each other in close corridors). In general, we want to use the MABS presented here to investigate questions that cannot be studied with classical discrete event simulations meaning questions that depend on individual decisions.

ACKNOWLEDGEMENTS

This work was partially founded by European Regional Development Fund (ERDF). A large part of the work presented in this paper was done by Carsten Beth and Wilm-Hendrik Scholz.

REFERENCES

Johnson, L. W. (1990), Discrete Choice Analysis with Order Alternatives, In M. M. Fisher, P. Nijkamp and Y. Y. Papageorgiou (eds), *Spatial Choices and Processes*, North Holland, Amsterdam.

Winston, W. L. (1993), *Operations Research Applications and Algorithms*, Duxbury, Belmont, California.

Castilho, D. B. and Daganzo, C. F. (1993), Handling Strategies for Import Containers at Marine Terminals, *Transportation Research*, l 27B, 151-166.

Bates, J.J. and Roberts, M. (1983), Recent Experience with models fitted to Stated Preference Data, Proceedings *11th PTRC Summer Annual Meeting*, University of Sussex, England, 4-7, July 1983.

Bangsow, S (2008): *Fertigungssimulationen mit Plant Simulation und SimTalk*. Hanser Verlag, München.

Becker, M.; Wenning, B.-L.; Görg, C.; Gehrke, J.D.; Lorenz, M.; Herzog, O. (2006): Agent-based and discrete event simulation of autonomous logistic processes. In: Borutzky, W.; Orsoni, A.; Zobel, R. (Hrsg.): *Proceedings of the 20th European Conference on Modelling and Simulation*. Sankt Augustin, Bonn.

Beth, C., Kamenik, J., Ommen, D., Hahn, A. (2011): Design aspects of cognitive logistic systems. In Kreowski, H.-J.; Scholz-Reiter, B.; Thoben, K.-D.: *Dynamics in Logistics*. Springer, Berlin.

Davidsson, P. (2001): Multi Agent Based Simulation: Beyond Social Simulation. In: Moss, S.; Davidsson, P. (Hrsg.): *Multi-Agent-Based Simulation*. Springer, Berlin.

Follert, G.; Roidl, M. (2008): Evaluation of Routing Strategies for Decentralized Self-Organisation in Large Scale Conveyor Systems. In: Kimberly, E. et. al. (Hrsg.): *Proceedings of the 10th International Material Handling Research Colloqium*. Material Handling Industry of America, Charlotte, USA.

Gehrke, J. D.; Lorenz, M.; Wenning, B.-L.; Becker, M. (2006): Integration of two approaches for simulation of autonomous logistic processes. In Wenzel, S. (Hrsg.): *Tagungsband zur 12. ASIM Fachtagung*. SCS Publishing House e.V., Erlangen.

Günthner, W.; ten Hompel, M. (Hrsg.) (2010): *Internet der Dinge in der Intralogistik* (VDI-Buch). Springer, Berlin.

Kuhn, A.; Wenzel, S (2007): Simulation logistischer Systeme. In: Arnold, D. et al. (Hrsg.), *Handbuch Logistik*. 3. Auflage. VDI Springer, Berlin Heidelberg.

Kummer, O.; Wienberg, F.; Duvigneau, M.; Schumacher, J.; Köhler, M.; Moldt, D.; Rölke, H.; Valk, R. (2004): An extensible editor and simulation engine for Petri nets: Renew. In: Cortadella, J.; Reisig, W. (Hrsg.): *25th International Conference on Application and Theory of Petri Nets* (ICATPN 2004), Springer, Berlin.

Li,Y; Oberweis, A; Mevius, M. (2006): Simulation von Supply Chains mit Petri-Netzen. In Becker, M; Szczerbicka, H. (Hrsg.), *Proceedings of the 19th Symposium on Simulationstechnique (ASIM'2006)*. SCS Publishing House e.V., Hannover.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G. (2005): Mason: A multiagent simulation environment. *Simulation* 81: 517–527

Ommen, D., Kamenik, J., Beth, C., Busch, J.C.; Kulas, A.; Cramer, E.; Hahn, A. (2010): An Adaptable Architecture for Intelligent Conveyors. In Filipe, J.; Andrade-Cetto, J.; Ferrier, J.-L. (Hrsg.): *ICINCO* (1). INSTICC Press.

Reisig, W.; Rozenberg, G. (Hrsg.) (1998): Lectures on Petri Nets I: Basic Models. *Lecture Notes in Computer Science* Vol. 1491, Springer, Berlin.

Scholz-Reiter, B.; Höhns, H. (2006): Selbststeuerung logistischer Prozesse mit Agentensystemen. In: Schuh, G. (Hrsg.), *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte* (VDI-Buch). Springer, Berlin.

Wenzel, S.; Noche, B. (2000): Simulationsinstrumente in Produktion und Logistik – eine Marktübersicht. In Mertins, K.; Rabe, M. (Hrsg.), *The New Simulation in Production and Logistics*: Prospects, Views and Attitudes. IPK Eigenverlag, Berlin.

Wenzel, S. (Hrsg.) (2006): *Simulation in Produktion und Logistik*. Tagungsband zur 12. ASIMFachtagung Simulation in Produktion und Logistik. SCS Publishing House e.V., Erlangen.