Computers & Industrial Engineering 126 (2018) 482–493

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

An integrated scheduling method for AGV routing in automated container terminals

Yongsheng Yang[a,*], Meisu Zhong[a], Yasser Dessouky[b], Octavian Postolache[c]

[a] Shanghai Maritime University, Shanghai 201306, China
[b] San Jose State University, USA
[c] Instituto de Telecomunicacões/ISCTE-IUL, Lisboa, Portugal

ARTICLE INFO

Keywords:
Automated container terminal
AGVs path planning
Container handling
Integrated scheduling

ABSTRACT

The simultaneous scheduling of quay cranes (QCs), automated guided vehicles (AGVs), and yard cranes (YCs) in automated container terminals (ACTs) has been a critical problem. This paper proposes an integrated scheduling for handling equipment coordination and AGV routing. With the goal of minimising makespan, we set up a bi-level programming model. To solve the model, we investigate and compare the rolling horizon procedure (RHP) and Congestion Prevention Rule-based Bi-level Genetic Algorithm (CPR-BGA). It is shown that the CPR-BGA algorithm is highly effective for the integrated scheduling in ACTs. We conclude that the CPR-BGA is effective.

1. Introduction

The continuous development of global trade as well as logistics technologies has been pushing up the demand for container terminals, including loading and unloading operations, and the storage area. Automated Container Terminals (ACTs) have appeared to meet this ever-increasing demand and contribute to the higher efficiency and productivity of port operations as well as reduction in the cost of human resources and emissions at a port. More than 20 ACTs have taken off around the world but how to continue to improve their efficiency is still one of the most frequently discussed topics on port operations and management.

The research in the literature on this topic can be categorized into three groups. One is concerned with the methodology of how to describe operations of an ACT. Liu, Jula, and Ioannou (2002) presented a setup and analysis of a microcosmic simulation model for the operation mode of AC. They showed that the model significantly improved the performance of the traditional terminal and reduced costs. Hu, Lee, Huang, Lee, and Chew (2013) considered an automated terminal system that was decomposed into three subsystems through the Markov chain model to analyse and forecast the capacity of containers in ACT. The sensitivity analysis illustrated the high efficiency of ACT in the latter study. Yang et al. (2015) analysed two kinds of microscopic parameter models to assess the performance of ACT based on automatic stacking cranes (ASCs) or AGVs. The implemented models revealed advantages, such as higher efficiency and stability, effectively reduced labour costs, decreased the probability of personal injury accidents, and proved to be safer for the terminal environment. Automated terminals possess many advantages in terms of labour cost, improved operation efficiency and economic benefit, reduced energy consumption, and improved levels of safe operation and the port's reputation (Le, Yassine, & Riadh, 2012; Zhang, Ioannou, & Chassiakos, 2006; Martín-Soberón, Monfort, Sapiña, Monterde, & Calduch, 2014). While an ACT offers such crucial benefits, more research should be conducted to further rationalize its use.

The operation mode of automated container terminal can be divided into a loading process and unloading process. In the loading process, containers from the storage location are transferred from the yard by YCs to AGVs, which then transport containers to the port to be loaded onto ships by QCs. In the unloading process, containers are removed by QCs and transferred to AGVs, which transport the containers to the YCs that place containers to the corresponding storage location in the yard (Gharehgozli, Roy, & Koster, 2016). The operation of container terminals is shown in Fig. 1. In this work, we considered actual circumstances in automated terminals where the automatic rail-mounted gantry (ARMG) unloads the container from AGVs to the AGV-mate in the front of a yard or from the AGV-mate to AGVs. The AGV-mate is an auxiliary equipment of AGVs that is used to reduce the waiting times associated with ARMG and AGV, which can optimise the cycle of container loading and unloading and improve the efficiency of ACT.

Due to the development of large containers and rising labour costs, many overseas and domestic researchers have investigated the

* Corresponding author.
E-mail address: yangys_smu@126.com (Y. Yang).

https://doi.org/10.1016/j.cie.2018.10.007
Received 12 July 2018; Received in revised form 2 October 2018; Accepted 4 October 2018
Available online 06 October 2018
0360-8352/

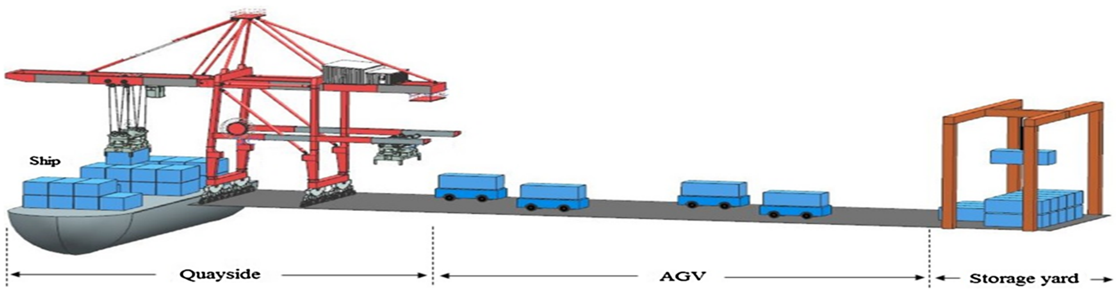

Fig. 1. The operation of container terminals.

scheduling problem of AGVs for decades. Since the container-throughput in Chinese ports have taken a great leap forward, the scheduling problem of AGVs has become more urgent. In many studies, AGVs and QCs are considered independently. However, AGVs and QCs work in a close relationship and should be studied dependently for two reasons: (1) AGVs play an important role in the quayside and yard-side operations, which restricts and influences the operation of the other two parts, and (2) the scheduling of AGVs affects the time needed for every container to be handled by AGVs. AGVs, as the carrier to connect QCs with YCs, can influence the operation system of QCs and YCs, which is directly related to the working efficiency of terminals, and affect the capacity of a container terminal. In addition, owing to the high cost of QCs or other factors like the quantitative restriction, an inefficient operation will cause delays in container terminals and increase the transportation cost of containers. Therefore, studying the problem of integrated scheduling of QCs, AGVs, and YCs is the key to automated terminal efficiency. Moreover, a series of problems still exists, including AGV congestion and conflict, especially in the practical operation of automated terminals. Considering that there are many indefinite factors in current running environments, realizing dynamic scheduling based on the conditions of AGV path planning is of great significance in solving the practical problems in automated terminal. In this work, we put forward integrated scheduling of QCs, AGVs, and ARMG and path planning for AGVs to reduce the problems of AGV conflict and congestion, which has yet to be done in previous literature. Therefore, we studied path planning to achieve the integrated scheduling of QCs and AGVs and the ARMG optimisation method in the automation terminal. We also propose a bi-level programming model and Congestion Prevention Rule-based Bi-level Genetic Algorithm (CPR-BGA).

This paper is structured as follows. Section 2 provides a review of the existing literature on AGV path planning and handling equipment scheduling. Section 3 presents a bi-level programming model formulated for the problem of interest. Section 4 proposes a bi-level Genetic Algorithm to solve the formulated model handling a large-sized problem. Section 5 shows and analyses the numerical results to show the performance of the formulated model and proposed algorithm. Section 6 presents the conclusion and offers suggestions for further research.

2. Literature review

In recent years, there has been numerous researches on the equipment of automation terminals, especially focused on improving the efficiency of QCs. For example, Meisel and Bierwirth (2009, 2013), combined berth allocation with QCs assignment, by the heuristic algorithm to get the solution, and then they build a framework with three stages to finish the integrated planning, which has improved the resource utilization and reduced costs in terminals. Park, Choe, Ok, and Ryu (2010) used the heuristic-based and local-search-based real-time scheduling methods and analysed the reason for the delay when operating QCs and AGVs to better increase QC utilization and reduce costs. Wen, Ekşioğlu, Greenwood, and Zhang (2010) also attempted to minimise the interference between the QCs and improve the utilization of QCs by dividing the task allocation of QCs through the Ant Colony Optimisation Algorithm to increase efficient scheduling of QC projects. Li et al. (2016) considered the interference among QCs and fixed the distance between the QCs, taking full account of the acceleration and deceleration of QCs in real-time operating. The latter research used heuristics and a rolling horizon algorithm to improve the efficiency and reduce the waiting time of QCs, which is a key to improve the throughput of container terminals.

While the above-mentioned methods merely focus on QCs in loading or unloading operation mode, we consider the actual port operation mode for loading and unloading of QCs in this work. Previous research that only focused on the QCs does not match the actual situation of automation terminals without considering the AGVs and the yards, which cannot improve the overall operating efficiency. The use of AGVs first appeared in 1950s, and now there are many researches on the AGV scheduling method. Grunow, Günther, and Lehmann (2006) first found that AGVs could handle two 20-ft containers or one large 40-ft container. The randomness of the off-line heuristic model was proposed to improve the efficiency of the AGVs. Nguyen and Kim (2009) discussed scheduling the automatic lifting vehicle (ALV) based on the time window constraints and analysed the number of ALVs and buffer capacity. Gupta, Roy, Koster, and Parhi (2017), using an integrated queuing network model to analyse the scheduling of ALVs and storage yard, find out that the parallel stack layout has a better performance. Pap, Bojanić, Georgijević, and Bojanić (2011) reduced the complexity of the container terminal operations and presented a new hypothesis of a simulation model based on the unloading mode and task allocation of containers to minimise the number of AGVs.

However, the above researchers do not consider QCs and AGVs scheduling simultaneously. When AGVs wait for a long time in the actual operation, congestion and conflict arise; thus, both QC and AGV scheduling should be examined together. No research presently exists on the optimal path planning, which greatly reduces the working efficiency of AGVs.

As mentioned before, there are numerous studies on AGVs independently, but they ignore the impact of QCs and the yard on AGVs. Meersmans and Wagelmans (2006) used the Branch and Bound algorithm with the goal of minimising the completion time to solve the scheduling problem of equipment in automated container terminals, which was the first attempt of integrated scheduling of AGVs, QCs, and YCs. Chen, Bostel, Dejax, Cai, and Xi (2007) set up a Hybrid Flow Shop Scheduling problem with precedence and Blocking constraints (HFSS-B) to analyse the integrated scheduling method of QCs and Yard Vehicles (YVs). Lau and Zhao (2008) used the multi-level genetic algorithm (MLGA) to solve the mixed integer programming model by reducing the berth time of ships and increasing the productivity of terminal to implement the effective scheduling of QCs, AGVs, and YCs. Liang, Lu, and Zhou (2009) analysed the unloading and loading mode to minimise the working time in order to take advantage of the heuristic algorithm based on Johnson's rule for scheduling. This algorithm implements the integrated scheduling of QCs, trucks, and YCs and has proven to be effective through the large-scale calculations. Wu, Luo, Zhang, and

Dong (2013) proposed the mixed integer programming (MIP) model with a minimum berth time and optimised the operating times of cranes through the non-linear mixed integer program (NIMIP) method to reduce the number of constraints and computing time. Skinner et al. (2013) put forward an optimisation method based on a genetic algorithm by comparing the different schemes used for integrated scheduling of QCs, trucks, and Straddle Carriers (SCs) to determine an optimised scheduling scheme. Dhingra, Roy, and Koster (2015), to achieve planning and scheduling of QCs and AGVs, combined the continuous-time Markov chain (CTMC) with the multi-class closed queuing network to get the solution of a two-level stochastic model, it proved that the vessel handling time was influenced by QCs operation and AGVs path.

Although much published literature exists on the scheduling of QCs, AGVs (or trucks), YCs, problems remain with determining the optimal route choice for path planning. According to the uncertainty of the actual operation, real-time dynamic planning has not been achieved, which is a highly important factor that influences the efficiency of automation terminals.

In regards to the AGV path planning problem, numerous studies have focused on the dynamic scheduling of AGVs, which motivated the basis of our study. For instance, Nishi, Ando, and Konishi (2006) realized the dynamic scheduling of AGVs by establishing 143 nodes of a transportation system to choose the optimal path of AGVs, which reduced lost time and improved scheduling efficiency. Su, Samuel, and Chong (2000) investigated AGV path planning as an NP-complete asymmetric traveling salesman problem with the minimum path as the goal. The researchers used an artificial neural network algorithm, known to be the basic characteristic of self-organizing, to solve the path planning of AGVs. Nishi, Hiranaka, and Grossmann (2011) minimised the delay time for finishing tasks as an objective by adopting the upper and lower levels of an algorithm for task allocation and focusing on scheduling as a higher level problem and path planning as a sub-problem. Results provided a feasible solution for AGV scheduling and avoided congestion of path planning at the same time. Liang, Lin, Gen and Chien (2012) considered the AGV in the flexible manufacturing system (FMS) by network model to implement the integrated scheduling and AGVs path planning using a random key-based particle swarm optimization (PSO) for crossover and mutation. Results proved that this algorithm was superior to the traditional genetic algorithm. Miyamoto and Inoue (2016) realized AGVs scheduling and conflict-free path planning by considering factors like the capacity constraints of vehicles, machine buffer, and so on, it is an integer programming problem, using the local and random search method to calculate. Sidoti et al. (2017) utilized the multi objective planning and asset routing (TMPLAR) tool and considered the uncertainty of space and time, resource constraints, and vehicle waiting for consumption. Roy, Gupta, and Koster (2016), proposed a solution for reducing the AGVs congestion in path planning and increasing the throughput capacity of ACTs, a non-linear traffic flow model was used to determine the appropriate number and effective speed of AGVs. Mishra, Roy, and Ommeren (2017), to solve the queuing problem in inter terminal transportation (ITT), using a network decomposition method to solve this problem, and applied the model to Port of Rotterdam. By doing so, they formed a time window for the shortest path based on the node network diagram to realize the AGV dynamic path planning.

In this work, we used the above-mentioned literatures on the optimal path selection of AGVs and equipment integrated scheduling as a basis to determine a practical solution for further optimisation and improved efficiency of automated terminals.

While numerous researches have published on the loading and unloading modes for vehicle scheduling and storage problems of containers in port yards, minimal studies exist on the integrated scheduling of QCs, AGVs, and ARMG in ACTs. For example, Zhang and Kim (2009) considered the loading and unloading mode to reduce the number of loading and unloading operations of QCs in two cycles. They created a container stacking sequence in the hatch in order using the local search method based on Johnson's rule, and the results demonstrated the validity of the optimisation. Zhang, Zeng, and Yang (2016) analysed the driving distance of trucks, YCs, and the required number of trucks and adopted the mixed intermediate storage policy. They combined the periodic queuing theory model, which minimises the driving distance of trucks and YC operation, improving the efficiency of the terminal. Luo and Wu (2015) achieved integrated scheduling of AGVs and YCs based on the loading and unloading mode and established an integer programming model by implementing the Genetic Algorithm for crossover and mutation; the results proved the effectiveness of the algorithm. Zhang, Jin, Ma, and Luan (2015) initially optimised the hatch of each berthing ship and the loading and unloading order of containers in hatch stacking by considering the loading and unloading mode to reduce QC operation time. Then, they optimised the cargo planning of export containers to further reduce the QC operation times, which was realised by using the Bi-level Genetic Algorithm to eventually prove the effectiveness of the scheduling method and improve the loading and unloading efficiency of the container terminal. Hu, Sheu and Luo (2016) investigated the background of the Yang Shan port in Shanghai to realise the integrated scheduling of the automatic stacker crane (ASC) and the automatic lifting vehicle (ALV). It was determined that full use of the yard space to improve the terminal's capacity, and practical application of the CPLEX and genetic algorithm was demonstrated. Roy and Koster (2015), developed the new integrated stochastic model based on the interactions among QCs, ALVs, and ASCs, using the iterative convergence algorithm to obtain reasonable methods and layouts of seaside operations.

As is evident, there are many studies on the integrated scheduling of QCs, trucks, and YCs in traditional container terminals (Boysen, 2010; Zhao and Tang, 2011; Boysen, Briskorn, & Tschöke, 2013), but they only focus on a single cycle of loading or unloading. Thus, the integrated scheduling of QCs, AGVs, and YCs for two or more cycles of loading and unloading need to be studied. Herein, considering the dynamic path planning of AGVs, we realised the integrated scheduling of QCs, AGVs, and ARMG simultaneously to develop a bi-level model for optimised planning.

3. Model formulation

The presented research focused on the loading and unloading operation modes with the aim to minimise the handling time of ships in port. The problem was divided into two parts: one as the integrated scheduling problem of AGVs and the second as the AGV path planning problem. When a ship first entered the docking area, we arranged for the QC and the block to be ahead of time for the ship. Then, the containers were discharged off the ship to the specified block, while another container was loaded onto the ship from the block at the same time. In order to improve the efficiency of loading and unloading, QCs need to match with blocks, which means that a designated QC unloads a container from a ship onto a specified block. The specific operation process for the loading and unloading mode is presented in Fig. 2, which shows that when the AGV receives an unloading or loading order, a container is discharged via the QC to a specified block or a container before the AGV-mate is transported to a specified QC.

In terms of unloading operations, the main trolley in the QC will first place containers on the transfer platform, then the portal trolley of the QC will transport a container from the transfer platform to the AGV. After that, the AGV will transport the container to a specified block of the AGV-mate, then the front ARMG of the yard will unload the container onto a stored block. In the following part of the sequence, the back ARMG moves the container from the stored block to a specified position in the yard, then the AGV receives the next task order at the same time. The loading operation is similar to unloading, but the operational process is the opposite. The horizontal transportation of AGVs follows the scheduling rules for the operation side, namely that the AGV can service any QCs and blocks. When the AGV transports the

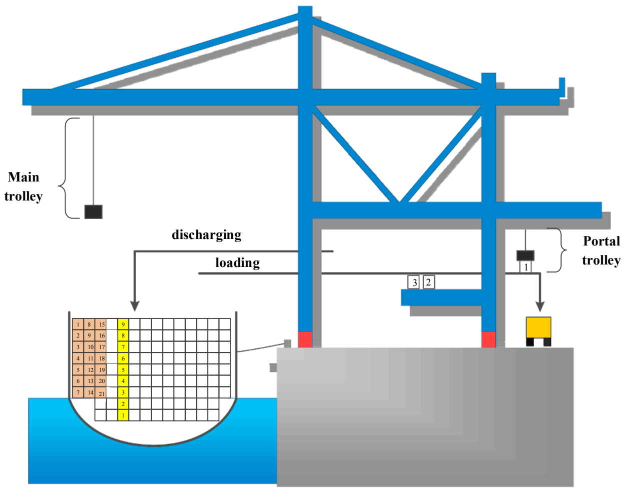

Fig. 2. The specific operation process of loading and unloading mode.

container, its driving optional path is established according to a set of rules. Based on these rules, a suitable path for horizontal transport is selected, and the process is repeated until all the containers have been loaded and unloaded.

Through the above problem, we know when containers are loaded and unloaded and the target position of the container, but which AGVs is needed to exactly transport the container is unknown, so we need to assign tasks for AGVs (Mishra and Tripathi, 2015). After the assigned tasks, we plan the path for AGVs, and the alternative driving path of AGV is known. According to the set rules of the auxiliary road, we then select an optimal path from alternative paths (Li, Caö, & Tan, 2011).

The layout of an automated container terminal is presented in Fig. 3, which indicates that the moving direction of the AGV is determined first in clockwise motion to reduce the waiting time and probability of congestion when planning the AGV path. Because the single circle line driving is low in efficiency, we choose to increase the auxiliary roads in the whole circle paths. The speed of AGVs is considered constant, therefore, the path planning problem can be transformed into the SPP-shortest path problem.

Horizontal transportation is crucial to link QCs and YCs when there are many containers in an automated container terminal, which will improve the loading and unloading efficiency and increase the quantity of AGVs to transport. When simultaneously using many AGVs in horizontal transportation, problems like traffic congestion and AGV conflict may appear. To solve the congestion problem of horizontal transportation, a rule should be set for vehicle density on the auxiliary road, which is defined as the number of AGVs on each road in the transportation area at the same time.

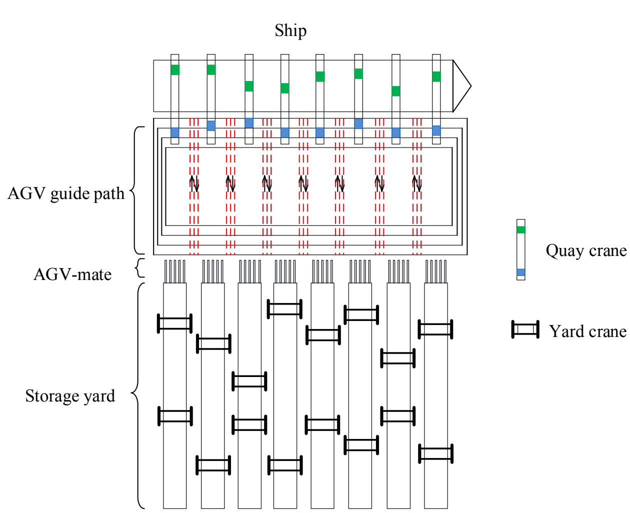

Fig. 3. The layout of the automated container terminal.

To reduce the probability of congestion, the two AGVs are allowed on each auxiliary road at the same time. If there are more than two, another path needs to be planned for the third AGV so there is no possibility of congestion. Fig. 4 shows our proposed model map, which uses nodes to represent the inflection points, the crossing points in horizontal transport, blocks, and QCs.

3.1. Assumptions

(1) The main trolley of the QC loads the container onto the transfer platform or unloads the container from the transfer platform onto the ship, regardless of the capacity problem of transfer platforms in QCs. The transfer platform can accommodate multiple containers at the same time. Its running time is associated with the location of containers on the ship, which is assumed to exhibit uniform distribution.
(2) The portal trolley of the QC unloads the container onto the AGV or places the container from the AGV onto the transfer platform, for which the running time has a fixed value.
(3) The speed of AGVs is uniform, regardless of the time AGVs spent on QCs and AGV-mates.
(4) All the containers are 40 ft in size, and the QC and ARMG can load and unload one container at a time. The AGV also can transport one container at a time.
(5) ARMG in front of the blocks gets the container from the AGV-mate, the time of which is negligible, while the time to discharge the container from the AGV-mate onto the storage area is a fixed value.
(6) The time required for ARMG in the back of the blocks to retrieve the container from the storage area can be ignored. The time needed to discharge the container from the storage area to a specific location in the back of the block follows uniform distribution, namely the running time within a time.
(7) A loading task is completed before a discharging task, or a container is discharged from a ship to complete a loading task.

3.2. Model parameters

1) Set

U	set of import containers, $(1, 2, 3......i) \in U$
L	set of export containers, $(1, 2, 3......j) \in L$
V	set of AGVs, $(1, 2, 3......c) \in \mathrm{V}$
B	set of blocks in yards, $(1, 2, 3......b) \in \mathrm{B}$
Q	set of QCs, $(1, 2, 3......k, l) \in N$
N	set of all containers, $U \cup L$
G	set of nodes at a cross-road in horizontal transportation
K	set of nodes at an uncross-road in horizontal transportation
N^*	$B \cup Q \cup G \cup K$
N_k	set of the QC k handling the containers
S	a dummy starting QC
F	a dummy ending QC
O_S	$Q \cup S$, set including all QCs (i.e. all the containers to be loaded and unloaded by QCs) plus the dummy starting QC
O_F	$Q \cup F$, set including all QCs plus the dummy ending QC
O	$Q \cup S \cup F$, set including all real QCs, the dummy starting QC and ending QC
A	set the directed line segment between the nodes

2) Parameter

v	The speed of AGVs
M	A very large positive number
T_1	

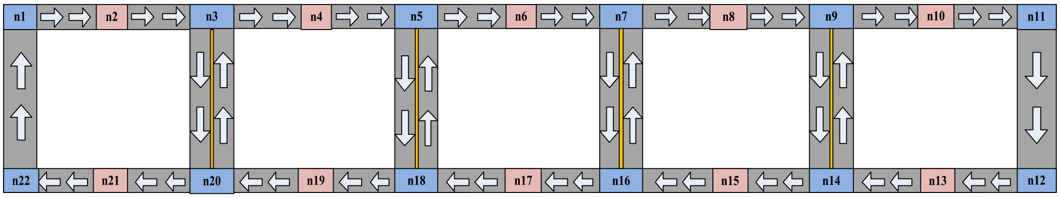

Fig. 4. The path planning of AGV.

The time of the portal trolley unloads the container to the transfer platform under the QC, also said the time of the container in apron be discharged to the transfer platform

T_2 The time of the first ARMG puts the container on the storage area

e_{ik} The time when the QC k uses the main trolley to handle the ith container from the ship to the transfer platform, also said the time when the QC k puts the ith container from the transfer platform to the ship

$t_{c,i'j'}$ The time of AGV c from node i' to node j'

g_{ik} The time when the second ARMG putting the i th container from the storage area to the end-position after the QC k handling the i th container

$d_{i'j'}$ The distance between node i' and node j'

$a_{i'j'}$ The connection matrix between node i' and node j'

$b_{i'j'}$ The auxiliary road of the connection matrix between node i' and node j'

$T_{v,i'j'}$ The time window function of node i' and node j'

3) Decision variables

No 0–1 variables

u_{ik} The time when the QC k starts to handle the ith container, $i \in N_k$

r_{ik} The time when the QC k uses the portal trolley to get the ith container from AGV or the ith container to be put on the AGV

h_{ik} The time when the QC k unloads the ith container on AGV-mate or loads the ith container from AGV-mate

f_{ik} The time when the QC k finishes the ith container

$t_{in,i'j'}$ The time when the AGV enters into the auxiliary road of node i' and node j'

$t_{out,i'j'}$ The time when the AGV leaves away the auxiliary road of node i' and node j'

0–1 variables

x_{ikjl} The AGV, which just handling the ith container of the QC k is scheduled to handle the jth container of QC l, $i \in U, j \in L$ or $i \in L, j \in U$

β_{ikc} The AGV c to handle the ith container of QC k

α_{ikb} the ith container of QC k is located in block b

$y_{i'j'}$ If the AGV gets through the connected path of node i', j'

$z_{i'j'}$ If the vehicle density of the auxiliary road of node i' and node j' is more than 2

3.3. Model 1: Scheduling model

The above assumptions and parameters are considered to build the model and minimise the makespan of loading and unloading the ship via bi-level programming.

$$min\ \mathrm{T} = \max_{k \in O}\{F_k - S_k\} \tag{1}$$

The objective of this model is to minimise the time difference between finishing the last container and starting the first container, which represents the loading and unloading of the makespan of the ship.

Subject to:

$$F_k = \max_{i \in N_k}\{f_{ik}\} \quad \forall\ k \in O_F \tag{2}$$

$$S_k = \min_{i \in N_k}\{r_{ik}\} \quad \forall\ k \in O_S \tag{3}$$

Constraint (2) means that we choose the latest time from a set of all the containers as the completion time of the last container. The constraint expressed on (3) ensures that we consider the earliest time from a set of all containers as the starting time of the first container.

$$\sum_{l \in O_F} \sum_{j \in L} x_{ikjl} = 1 \quad \forall\ k \in O_S, \forall\ i \in U \tag{4}$$

$$\sum_{k \in O_S} \sum_{i \in U} x_{ikjl} = 1 \quad \forall\ l \in O_F, \forall\ j \in L \tag{5}$$

$$\sum_{i \in N_k} \beta_{ikc} = 1 \quad \forall\ k \in O_S, \forall\ c \in V \tag{6}$$

$$\sum_{i \in N_k} \beta_{ikc} = 1 \quad \forall\ k \in O_F, \forall\ c \in V \tag{7}$$

$$\sum_{i \in N_k} \alpha_{ikb} = 1 \quad \forall\ k \in O_S, \forall\ b \in B \tag{8}$$

$$\sum_{b \in B} \alpha_{ikb} = 1 \quad \forall\ i \in N_k, \forall\ k \in O_S \tag{9}$$

Constraint expressed on (4) ensures that the same AGV complete a loading task using a QC only after finishing an unloading task. Constraint (5) implies that after the QC finishes a loading task, the same AGV can only complete an unloading task by the QC. Constraint (4) and Constraint (5) confirm completion of the loading and unloading processes. Constraint (6) ensures that each loading and unloading task is handled only by one AGV. Constraint (7) implies that every AGV only handles one container at a time. Constraint (8) means that the ARMG transports a container to be stacked in the assigned block. Constraint (9) means that each block can only load and unload one container at a time via the ARMG.

$$u_{ik} + e_{ik} + T_1 \leqslant r_{ik} \quad \forall\ i \in U_k, \forall\ k \in O \tag{10}$$

$$r_{ik} + \sum_{c \in V} t_{c,i'j'}\beta_{ikc} \leqslant h_{ik} \quad \forall\ i \in U_k, \forall\ k \in O, \forall\ i' \in O, \forall\ j' \in B \tag{11}$$

$$h_{ik} + T_2 + g_{ik} \sum_{b \in B} \alpha_{ikb} \leqslant f_{ik} \quad \forall\ i \in U_k, \forall\ k \in O \tag{12}$$

Constraints (10)–(12) represent that the relationship between each time that a ship starts and finishes a discharging task. Constraint (10) represents the time that the AGV begins to transport the container from the QC based on the time that the portal trolley in the QC handles the container. Constraint (11) represents the relationship between the time that the AGV starts to transport the container to the AGV-mate. Constraint (12) signifies the relationship between the time the container is finished by ARMG and the time it is transported to the AGV-mate.

$$h_{ik} + \sum_{c \in V} t_{c,i'j'}\beta_{ikc} \leqslant h_{jl} + \mathrm{M}(1 - x_{ikjl}) \quad \forall\ i \in U_k, \forall\ j \in L_l, \forall\ k \in O_S, \forall\ l \in O_F, \forall\ i', j' \in B \tag{13}$$

Constraint (13) gives the relationship between the time that the same

AGV finishes an unloading task and starts the next loading task.

$$u_{jl} + g_{jl} \sum_{b \in B} \alpha_{jlb} + T_2 \leqslant h_{jl} \quad \forall\, j \in L_l, \forall\, l \in O \tag{14}$$

$$h_{jl} + \sum_{c \in V} t_{c,i'j'} \beta_{jlc} \leqslant r_{jl} \quad \forall\, j \in L_l, \forall\, l \in O, \forall\, i' \in B, \forall\, j' \in Q \tag{15}$$

$$r_{jl} + T_1 + e_{jl} \leqslant f_{jl} \quad \forall\, j \in L_l, \forall\, l \in O_F \tag{16}$$

Constraints (14)–(16) express the relationship between the time the container is initially loaded from the block and completion of loading at each time of shipment. Constraint (14) represents the time that the ARMG in the back of the block gets the container, which is less than the time needed for AGV to obtain the container from the AGV-mate. Constraint (15) means that the starting time of AGV from the block does not exceed the time the AGV arrives at the QC. Constraint (16) implies that the time the portal trolley in the QC obtains the container from the AGV is no more than the time that the loading task is finished.

$$r_{jl} + \sum_{c \in V} t_{c,i'j'} \beta_{jlc} \leqslant r_{ik} + \mathrm{M}(1 - x_{jlik}) \quad \forall\, i \in U_k, \forall\, j \in L_l, \forall\, k \in O_S, \forall\, l \in O_F, \forall\, i', j' \in Q \tag{17}$$

Constraint (17) represents the time relation between completion of a loading task by the AGV and start of the next unloading task.

$$u_{(i+1)k} - u_{ik} = e_{ik} + e_{(i+1)k} \quad \forall\, i \in U_k, \forall\, k \in O \tag{18}$$

$$u_{(i+1)k} - u_{ik} = g_{ik} + g_{(i+1)k} \quad \forall\, i \in L_k, \forall\, k \in O \tag{19}$$

$$u_{ik} \geqslant 0, r_{ik} > 0, h_{ik} > 0, f_{ik} > 0, g_{ik} > 0, e_{ik} > 0 \quad \forall\, i \in N_k, \forall\, k \in O \tag{20}$$

$$t_{c,i'j'} \geqslant 0 \quad \forall\, i', j' \in N^*, c \in V \tag{21}$$

Constraint (18) explains the time relation of the main trolley in the QC when it starts unloading two consecutive containers. Constraint (19) represents the time relationship of ARMG in the back of the block when it loads two consecutive containers. Constraint (20) represents the range of time parameters. Constraint (21) expresses the range of time parameters of the AGVs' transportation.

3.4. Model 2: Path planning model of AGVs

To determine the time for AGV transportation in the path planning model in this work, we obtained the minimum makespan when all the containers were finished. If the AGV transportation time is short, the loading and unloading makespan are also short. According to Fig. 4, the model can be described by the following equation:

$$t_{c,i'j'} = \min\left(\sum_{(i',j') \in A} y_{i'j'} (d_{i'j'} + M(1 - z_{i'j'})) \right) / v \quad \forall\, i', j' \in N^*, c \in V \tag{22}$$

Constraint (22) represents the time of the AGV's horizontal transportation with constant speed. We use the length of the AGV moving path divided by the AGV's speed to represent the time of horizontal transportation.

$$\sum_{j':(i',j') \in A} y_{i'j'} - \sum_{j':(i',j') \in A} y_{j'i'} = \begin{cases} 1, & i' = s, \\ -1, & i' = t, \\ 0, & i' \neq s, t, \end{cases} \quad \forall\, i', j' \in N^* \tag{23}$$

$$y_{i'j'} = \begin{cases} 1, & \textit{the AGV runs this road,} \\ 0, & \textit{otherwise.} \end{cases} \quad \forall\, i', j' \in N^* \tag{24}$$

$$y_{i'j'} \leqslant a_{i'j'} \quad \forall\, i', j' \in N^* \tag{25}$$

$$z_{i'j'} = \begin{cases} 1, & \textit{If the density of auxiliary vehicle is less than 2,} \\ 0, & \textit{If the density of auxiliary vehicle is 2.} \\ \forall\, i', j' \in G \end{cases} \tag{26}$$

$$z_{i'j'} \leqslant b_{i'j'} \quad \forall\, i', j' \in G \tag{27}$$

From constraints (23)–(27), we obtain the constraints for the driving distance of AGVs. Among them, constraint (23) represents the distance between the node i' to node j, which are equal. Constraint (24) shows that the AGV will get through the route of node i' and node j. Constraint (25) gives the relationship between the 0–1 variable and the connection matrix. Constraint (26) shows that the AGV can get through the auxiliary road of node i' and node j'. Constraint (27) provides the relationship between the 0–1 variable and the connection matrix of the auxiliary road.

$$T_{V,i'j'} = (c, t_{in,i'j'}, t_{out,i'j'}) \quad \forall\, i', j' \in G \tag{28}$$

$$t_{out,i'j'} = t_{in,i'j'} + d_{i'j'}/v \quad \forall\, i', j' \in G \tag{29}$$

$$\left(t_{in,i'j'}\right)_c < \left(t_{in,i'j'}\right)_{c+1} \quad \forall\, i', j' \in G \tag{30}$$

$$\left(t_{in,i'j'}\right)_c + d_{i'j'}/v \leqslant \left(t_{in,i'j'}\right)_{c+2} \quad \forall\, i', j' \in G \tag{31}$$

Constraints (28)–(31) are window of time constraints, the time window constraints that are used to judge the vehicle density of the auxiliary road and then determine if the AGV can get through the road. Constraint (28), the time window, concerns the time that the AGV enters and leaves the auxiliary road. Constraint (29) represents that the time relationship of the AGV entering and leaving the auxiliary road. Constraint (30) shares the time relation of two consecutive AGVs when they enter the same auxiliary road. Constraint (31) provides the necessary conditions at the time the first AGV enters the auxiliary road, which should not exceed the time that the third AGV enters the auxiliary road.

$$r_{ik} + \sum_{c \in V} (d_{ki'}/v) \beta_{ikc} = t_{in,i'j'} \quad \forall\, i', j' \in G, \forall\, i \in U_k, \forall\, k \in Q \tag{32}$$

$$t_{out,i'j'} + \sum_{c \in V} (d_{j'b}/v) \beta_{ikc} \leqslant h_{ik} \quad \forall\, i', j' \in G, \forall\, k \in Q, \forall\, i \in U_k, \forall\, b \in B \tag{33}$$

Constraint (32) explains that the time of the AGV starting to handle the discharging container by the QC and AGV from the QC to auxiliary road is the time of AGV entering into the auxiliary road. Constraint (33) ensures that the time the container is discharged onto the AGV-mate does not exceed the time the AGV leaves the auxiliary road.

Constraint (32) explains that the time of the AGV starting to handle the discharging container by the QC and AGV from the QC to auxiliary road is the time of AGV entering into the auxiliary road. Constraint (33) ensures that the time the container is discharged onto the AGV-mate does not exceed the time the AGV leaves the auxiliary road.

$$h_{ik} + \sum_{c \in V} (d_{ab}/v) \beta_{ikc} \leqslant h_{jl} + M(1 - x_{ikjl}) \quad \forall\, i \in U_k, \forall\, j \in L_l, \forall\, a, b \in B, \forall\, k \in O \tag{34}$$

$$h_{jl} + \sum_{c \in V} (d_{bi'}/v) \beta_{jlc} = t_{in,i'j'} \quad \forall\, i', j' \in G, j \in L_l, l \in Q, b \in B \tag{35}$$

$$t_{out,i'j'} + \sum_{c \in V} (d_{j'b}/v) \beta_{jlc} \leqslant r_{jl} \quad \forall\, i', j' \in G, j \in L_l, l \in Q, b \in B \tag{36}$$

$$r_{jl} + \sum_{c \in V} (d_{lk}/v) \beta_{jlc} \leqslant r_{ik} + M(1 - x_{jlik}) \quad \forall\, i \in U, \forall\, j \in L_l, \forall\, l, k \in Q \tag{37}$$

Constraint (34) represents the relationship of the time for the AGV to complete the discharging task and begin the next loading task. Constraint (35) expresses that the time the AGV starts to unload is equal to the time AGV enters the auxiliary road. Constraint (36) ensures that the time the AGV finishes the loading task is no more than the time it takes to leave the auxiliary road. Constraint (37) represents the time relationship between the AGV finishing the loading task and starting the next unloading task.

The integrated scheduling problem of AGVs and QCs, which is based on path planning is a nested model, and no accurate software exists to solve the problem, especially for large-size loading and unloading modes. Therefore, in the next section, we present the Bi-level Genetic Algorithm as a solution for this problem.

4. Algorithm

We combined a heuristic algorithm with Genetic Algorithm (Harik, Lobo, & Goldberg, 2002; Baker and Ayechew, 2003; Pezzella, Morganti, & Ciaschetti, 2008), and provided reasons for this approach as follows: (1) It can be very effective in global search, and its great ability of obtaining accurate solutions. (2) Its flexibility, expansibility can be integrated can to the heuristic problem that was defined. (3) The proposed model is complex, which is a multiple objective function and constraint, and the heuristic GA is suitable for solving this kind of problem (Chiang, 2005; Qu, Xing, & Alexander, 2013). In terms of heuristic GA in which the speed of convergence is fast and optimisation has high efficiency. we used the Bi-level Genetic Algorithm, to put the optimal solution of lower level as the decision variable of upper level, finally to achieve the optimal solution of the objective function (Yang, Zhang, He, & Yang, 2009; Wang, Wan, Wang, & Lv, 2008).

In this work, we considered the integrated scheduling of QCs, AGVs, and to construct the bi-level programming model. The upper level of the model concerns the integrated scheduling, while the lower level pertains to the path planning of AGVs. We aim to decrease or eliminate the phenomenon of AGV congestion in actual operations of automation terminals, which reduces the work efficiency of the terminal. As a solution, we added a rule to the algorithm to prevent congestion, the Congestion Prevention Rule-based Bi-level Genetic Algorithm, which is described in Fig. 5.

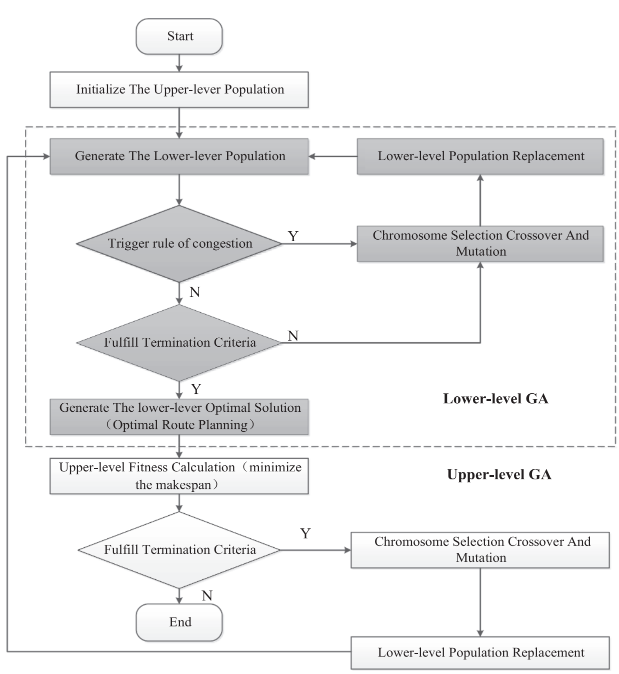

Fig. 5. The implemented genetic algorithm flowchart.

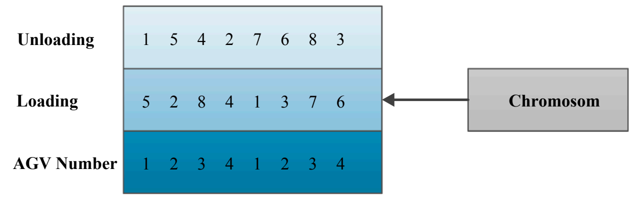

Fig. 6. Chromosome representation example for task.

4.1. Chromosome coding and encoding

1) The upper level coding

First, for the upper level of the model, we coded the QCs to load and unload containers and the task allocation of AGVs with integers. In order to better distinguish the loading and unloading of containers of QCs, we implemented multi-layer chromosomes. Because the QCs and blocks are matched in the assumed model, the loading and unloading of containers onto blocks is confirmed when the loading and unloading of containers of QCs is verified. At the time of AGV transportation and to reduce the driving distance of empty AGVs, the AGV will load a container once another container has been unloaded. After one loading is complete, the AGV will transport the next unloading container.

Fig. 6 illustrates the encoding process, assuming there are 2 QCs, each having 4 loading containers and 4 unloading containers, and 4 AGVs to transport. Numbers 1–4 represent the containers belonging to QC1, and numbers 5–8 represent the containers belonging to QC2. The number of AGVs is from 1 to 4, where AGV 1 and AGV2 perform the unloading task, and AGV3 and AGV 4 perform the loading task initially. The specific task allocation of AGV is presented in Fig. 7.

2) The lower level coding

The lower level of the model is for path planning of AGVs, for which we encode the path. The order of tasks of AGVs can be determined by the upper chromosomes, which confirms the order of the starting and ending point of the AGV performing the task. According to Fig. 4 in the last section, we know the node of AGVs to transport the containers. Taking AGV 1 in Fig. 6 as an example, AGV 1 first unloads the container of QC 1 and places it onto block 1. In other words, the starting node is 2 and the ending node is 8, as presented in Fig. 8 that shows the particular coding.

In this paper, chromosomes are decoded based on the priority weight, which allows us to plan the path for the AGV to unload the container from QC 1, we can plan the path of the AGV to unload the container from the QC 1. The AGV starts at node 2, goes through node 3, then through node 4 or 7, depending on their priority weights. In this

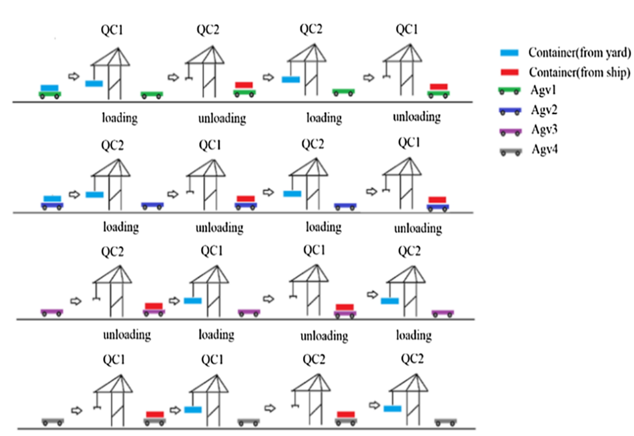

Fig. 7. The task allocation of AGV.

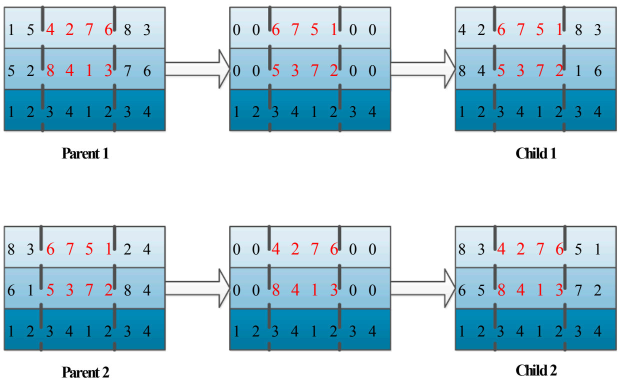

Fig. 8. Chromosome representation example for driving.

example, because node 7 has priority over node 4, AGV goes to node 7 and then finally to node 8, so the driving path of the AGV is 2-3-7-8.

4.2. Crossover and mutation

1) The upper level

Because AGVs can perform any loading and unloading tasks of QCs, no crossover for AGVs is considered, but crossover for the chromosome of the loading and unloading tasks is considered. In order to improve the efficiency of the crossover, Multi-point Crossover was used. The algorithm randomly selects multiple crossing points on chromosomes, where every two crossing points matches at a location and then chromosomes are exchanged between two points. Although each number can only appear once in the loading and unloading task, the exchange of chromosomes between two crossing points may lead to some code numbers to appear twice, so we may need to repair one of those chromosomes. First, 0 is used to complete the no-crossing chromosomes before crossover, so the separated chromosomes are added to the occupied positions of 0. If repeating numbers appear, then give it up and continue, until all the occupied positions of 0 have been replaced. The details are presented in Fig. 9.

Fig. 10 shows the process of swap mutation. The AGV needs to load a container (unload a container) after unloading a container (loading a container), so the mutation of the chromosome only needs to mutate the unloading or loading tasks. We adopted the Multi-points Mutation to randomly select the multiple points of mutation in a chromosome to form a new chromosome.

2) The lower level

A set of 34 nodes were considered, which has a relatively less number of nodes in the AGV path, and the Two-point crossover was adopted. At first, two crossing points were randomly generated, then two chromosomes were crossed to obtain chromosomes of the new offspring. This method of mutation is considered two-point mutation, where two points are randomly generated in a priority layer of chromosomes, then the two priorities are regenerated to replace the mutation point in the substring of chromosomes.

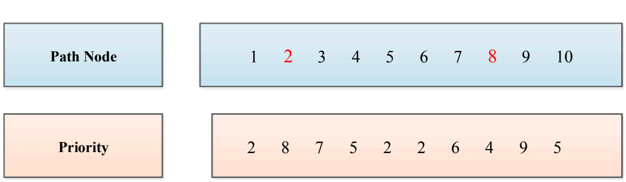

Fig. 9. An illustration of multi-point crossover for chromosome.

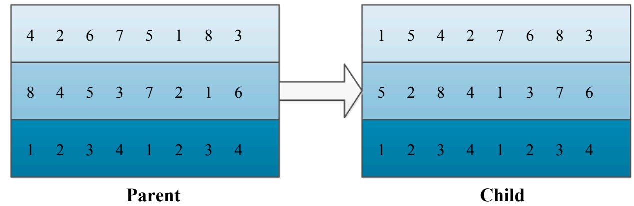

Fig. 10. An illusion of swap mutation for chromosome.

4.3. Stopping criterion

In order to balance the searching computation time and approximate an optimal solution, two stopping criterions were used: (1) the maximum number of evolving generations allowed for GA, which is a common criterion adopted by many GA-based optimisation problems. (2) The standard deviation of the fitness value of chromosomes (σL) in the current generation, which is below a small value (Baker & Ayechew, 2003). This parameter implies the diversity of the current generation in terms of the values of objective functions. The decreasing σL represents the decreasing diversity, where the algorithm is stopped if σL decreases below a small arbitrary constant ξ.

5. Numerical experiments

The proposed model and algorithm are validated for small and large-size problems. For small-size problems, the Rolling Horizon Procedure (RHP) (Vecchia, Marco, & Jean-Marie, 2012; Díaz-Madroñero, Mula, Jiménez, & Peidro, 2017) and Congestion Prevention Rule-based Bi-level Genetic Algorithm (CPR-BGA) were considered, and the objective functions values (OFV) and computation times were compared. However, the RHP method cannot be used to solve large-size problems within a reasonable computing time, so the proposed CPR-BGA algorithm was used to solve instead, which reduces the computing time and allows access of the OFV values. This Bi-level Genetic Algorithm was implemented in MATLAB 2016a, and all the simulations were performed on a computer with Intel Core TM i5 CPU 2.4 GHz and 8 GB RAM under a Windows operatingsystem. In order to reduce the deviation caused by the randomness of the Genetic Algorithm, every problem was solved 20 times, where the average computation time and OFV were used as the final results.

5.1. Parameter settings

(1) The considered number of loading and unloading containers varied from 1 to 2000, where 4–20 were considered for small-size problems and 30–2000 for large-size problems. We also considered the number of QCs and blocks in the range 2–8, while the number of considered AGVs varied from 8 to 20.
(2) The processing time of the main trolley that placed the container onto the transfer platform followed uniform distribution U (20, 40) s. The processing time of the portal trolley that placed the container onto the AGV from the transfer platform was fixed at 20 s, and the processing time of the ARMG localized in the front of the yard, which obtained the container from the AGV-mate and then unloaded it in the storage area, was fixed at 25 s. The processing time of ARMG localized in the back of the yard that moved the container from the storage area to the assigned truck followed uniform distribution U (20, 30) s. All of these values correspond with a real-time situation.
(3) In this work, we obtained port operation values from the Xiamen Ocean gate automated container terminal, which included an AGV horizontal speed of 5 m/s, AGV transportation length of 240 m, AGV transportation width of 100 m, and a distance between the adjacent auxiliary roads of 30 m.

Table 1
Results of computational experiments in small sizes.

No.	Containers	AGVs-QCs-Block	RHP		CPR-BGA		OFV gap rate (%)
			Computation time (s)	OFV (s)	Computation time (s)	OFV (s)	
1	4	2-2-2	119.37	143 s	3.46	157 s	8.91
2	8	4-2-2	221.56	203 s	2.74	223 s	8.97
3	8	6-2-2	208.90	189 s	3.08	202 s	6.44
4	10	3-3-3	404.67	218 s	1.89	241 s	9.54
5	12	6-3-3	444.89	164 s	2.52	170 s	3.53
6	12	9-3-3	373.78	154 s	2.85	162 s	4.94
7	16	4-4-4	528.43	280 s	1.76	302 s	7.28
8	16	8-4-4	612.28	207 s	1.95	217 s	4.61
9	24	12-4-4	934.82	183 s	2.17	194 s	5.67
10	30	5-5-5	/	/	1.83	403 s	/
11	30	10-5-5	/	/	2.64	325 s	/
12	30	15-5-5	/	/	2.98	302 s	/

(4) GA parameters were set based on preliminary tests, including a crossover rate (Pc) of 0.8, mutation rate (Pm) of 0.01, population size (Ps) of 20, and maximum generation (Mg) of 500.

5.2. Results for small-sized problems

Twelve small-sized experiments were performed, where the number of containers varied from 4 to 30. Table 1 shows that for the small-sized problems, the Bi-level Genetic Algorithm based on the rule of preventing congestion obtained better performances compared to the RHP algorithm in terms of speed, where computation time of the former algorithm ranged from 1.89 s to 3.46 s and that of RHP ranged from 119.37 s to 934.82 s. The RHP requires a rolling time axis for calculations, which increases the computation time. In addition, we observed that the difference of OFV between the Bi-level Genetic Algorithm based on the rule of preventing congestion and the RHP was small, where the average gap rate of OFV was 6.65%. The results also confirm that RHP cannot solve the large-size problems within a reasonable time frame. However, results were not obtained for experiments with more than 24 containers. Table 1 also shows that OFV exhibits the same growth trend as the number of containers increased.

5.3. Results for large-sized problems

From our experiments in Table 2, we conclude that using the congestion prevention rule of Bi-level Genetic Algorithm, can obtain a better OFV within a reasonable computation time. For the 39th case with 2000 containers, the computation time was 833.26 s, which shows that the CPR-BGA reduces the computation time. Under the same conditions, increasing the number of containers also increases the objective function value (completion of working time). For example, for the 21–22 case and 16–17 case, due to the multiple number of containers, the OFVs (finished working time) were similar. Under the condition with an invariant number of containers, an increased number of AGVs leads to a smaller OFV, namely the speed to finish the work is faster, such as in cases 14, 15, 19, and 20. Experiment results show that the terminal's work efficiency can be improved by adjusting the amount of AGVs, QCs, and ARMG. As shown in Table 2 for the cases with 32 containers, the OFV can be improved two- and threefold by increasing the number of AGVs from 13 to 14 and 15 and from 18 to 19 and 20, respectively.

It can also be observed that if will be duplicate the number of AGVs or even triplicate the number of AGVs, the values of objective functions become smaller, meaning enhanced efficiency. However, because the difference between OFV values of two times the number of AGVs and three times the number of AGVs is very little, we then comprehensively considered the cost factor in actual operations, where duplicating the number of AGVs proved to be more suitable in the model. From the

Table 2
Results of large-sized problems.

No.	Containers	AGVs-QCs-block	Computation time (s)	OFV (s)
13	32	2-2-2	2.54	891
14	32	4-2-2	3.03	498
15	32	6-2-2	7.09	398
16	64	4-2-2	9.03	969
17	120	4-2-2	28.25	1738
18	32	3-3-3	11.37	572
19	32	6-3-3	11.72	291
20	32	9-3-3	12.48	221
21	64	6-3-3	23.31	650
22	120	6-3-3	19.33	1253
23	64	4-4-4	4.10	949
24	64	8-4-4	5.18	561
25	64	12-4-4	13.34	488
26	240	8-4-4	55.19	1862
27	120	5-5-5	9.15	1442
28	120	10-5-5	11.99	810
29	240	10-5-5	44.02	1522
30	240	6-6-6	27.05	2347
31	240	12-6-6	37.76	1283
32	320	12-6-6	96.67	1642
33	320	8-8-8	35.70	2377
34	320	16-8-8	49.03	1286
35	400	8-8-8	53.70	2923
36	400	16-8-8	114.56	1427
37	800	18-9-9	269.46	2679
38	1000	20-10-10	255.68	3064
39	2000	20-10-10	833.26	5901

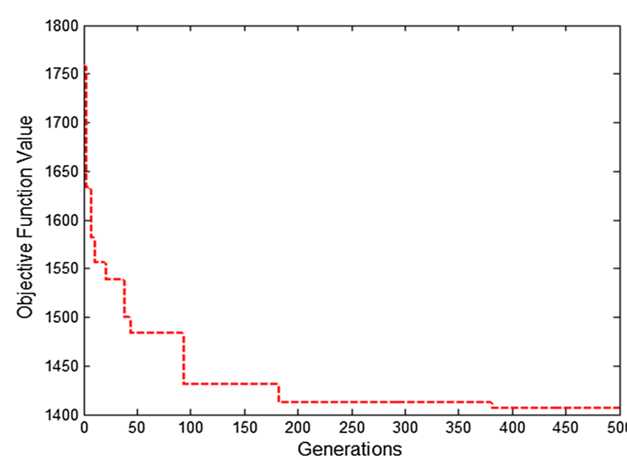

Fig. 11. Typical convergence of Bi-level Genetic Algorithm for case with 400 containers, 16 AGVs, 8 QCs and 8 Blocks.

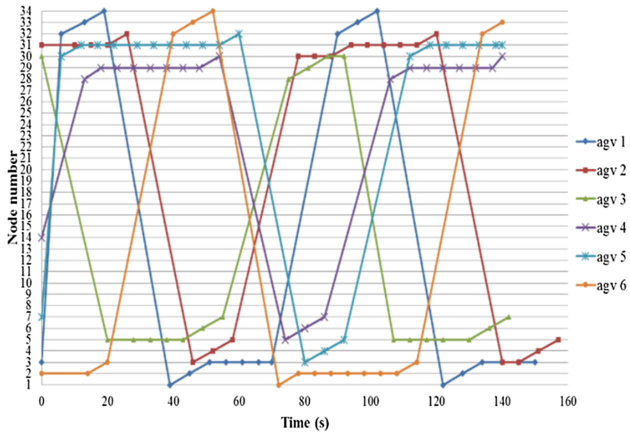

Fig. 12. Detailed routing result for the example of 12 containers, 6 AGVs, and 6 blocks without considering integrated scheduling and congestion prevention.

above discussion, it proved that choosing the appropriate number of QCs and AGVs, which is calculated using the model, is crucial in practice.

Fig. 11 shows the convergence of the Bi-level Genetic Algorithm for the case of 400 containers, 16 AGVs, 8 QCs, and 8 blocks. It reveals that the convergence of the Bi-level Genetic algorithm is relatively stable, which fully ensures the effectiveness of our designed algorithm. It also suggests that the convergence speed is high, which means that the results were obtained in a short time. A faster working time is highly important for the actual operation of container terminals because it relates to a higher efficiency of the working terminal.

By not considering the integrated scheduling of QCs and ARMG or the preventing congestion rule in path planning of AGVs, the routing nodes will be randomly assigned, and the choice for the optimal path is based on the Open Shortest Path First, which means the waiting time will be random and will decrease efficiency. Furthermore, by disregarding prevention of congestion problems, the AGVs will face congestion. This issue is seen in Fig. 12, where the congestion problem appeared for AGV 2 and AGV 5 within 10–20 s in node 31 and within 140–145 s in node 3. Further, it can be seen that the conflict problem occurred within 85 s when AGV 2 and AGV 3 were in node 31.

Fig. 13 displays case 5 from Table 1 for completing the process of loading and unloading, which shows the AGV path, task allocation, and time of operation, there are a total of 6 unloading containers, 6 loading containers, 3 blocks, 3 QCs to perform the loading and unloading, and 6 AGVs for the transport. During the first long period 0–50 s, the AGVs waited in a designated area before handling the first container. The second time period was when AGVs finished the last task and waited the start the next task. The driving process and task allocation of AGV 6 was designated as beginning to handle the task of block 1 and then performing the discharge task of QC 1. From this sequence, we see that the path of AGV 6 was: node 33 to node 34 to node 1 to node 2 to node 3 to node 32 and finally to node 33. The time that the AGV completed the first loading task was 72 s, while the second unloading task spent 140 s.

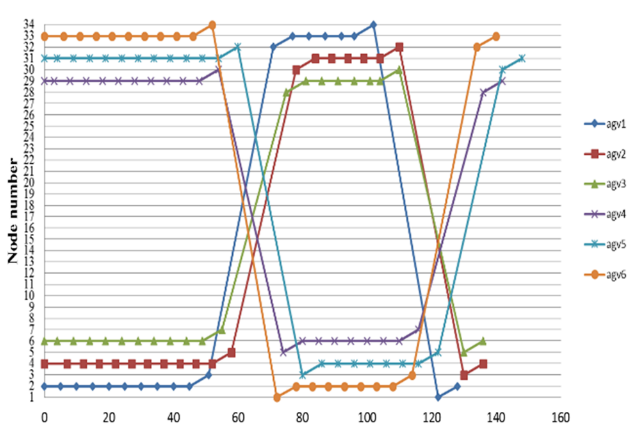

Fig. 13. Detailed routing result for the example of 12 containers, 6 AGVs, and 6 blocks considering integrated scheduling and congestion prevention.

By comparing Figs. 12 and 13, it can be concluded that the integrated scheduling of QCs, AGVs and ARMG and the introduction of the proposed preventive congestion rule for path planning resulted in a higher efficiency than the case without this rule. Fig. 13 reveals that only one AGV can be at a node at a time, or two AGVs cannot be at the same node simultaneously, which prevents conflicts and optimises the congested path. This satisfies all the timing constraints related to integrated scheduling and illustrates that the path planning method based on the rule of preventing congestion is successful and effective.

6. Concluding remarks

This paper has investigated the problem of integrated scheduling of QCs, AGVs and ARMGs for simultaneous loading and unloading operations, which has been formulated as a bi-level programming model. In this model, AGV path planning is considered at the lower level while the integrated scheduling of QCs, AGVs, and ARMGs is captured at the upper level with an aim to minimise the completion time of loading and unloading to improve the capacity of automated terminals.

A mixed integer programming model was used to solve conflictive problems and congestion to increase the suitability of the loading and unloading operation mode. However, only small-sized problems can be solved through the RHP algorithm, as computing time rapidly increases as the number of containers gets large. Therefore, to solve large-sized problems, or scheduling issues of many containers, we propose a bi-level General Algorithm based on the preventive congestion rule (CPR-BGA) to simulate actual situations of automated terminals that handle numerous containers. The numerical results suggest that the CPR-BGA algorithm can reduce the computing time, which is faster than using the RHP algorithm, and obtain better results within a reasonable time. The average gap between the RHP and CPR-BGA algorithms in terms of the objective function value for small-sized problems is very small but differs greatly for large-sized problems.

One key intellectual contribution of the paper is, from an academic perspective, we analysed the integrated scheduling of AGVs, QCs, and YCs to optimise path planning in automated terminals by simultaneously implementing the congestion prevention rule and the GA algorithm. Second, from a practical perspective, we consider the reality of loading and unloading in automated container terminals to achieve the optimised path and overcome the traffic bottleneck that leads to large-scale congestion in a long queue of AGVs. It is suggested that the experimental findings are beneficial for terminal managers who oversee daily operations.

However, as the number of large container ships increase, more effective scheduling strategies and decision-making algorithms will be needed for real-time scheduling and dynamic path planning. To further develop the proposed integrated scheduling model of QCs, AGVs, and YCs with the ability to assign containers to specific locations, more accurate and heuristic algorithms should be investigated in future studies. In addition, we can compare differemt algorithms with BGAs to see whether better results can be obtained, due to the complexity of the problem itself, and the GPU parallel computing can contribute to the algorithm to simplify the problem and reduce the computing time.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61540045), Shanghai Science and Technology Commission (No. 18295801100; 2018IB022; 16040501500; 17595810300).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.cie.2018.10.007.

References

Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research, 30*(5), 787–800.

Boysen, N., Briskorn, D., & Tschöke, M. (2013). Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectrum, 35*(2), 479–504.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research, 37*(1), 32–41.

Chen, L., Bostel, N., Dejax, P., Cai, J., & Xi, L. (2007). A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research, 181*(1), 40–58.

Chiang, C. L. (2005). Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE Transactions on Power Systems, 20*(4), 1690–1699.

Díaz-Madroñero, M., Mula, J., Jiménez, M., & Peidro, D. (2017). A rolling horizon approach for material requirement planning under fuzzy lead times. *International Journal of Production Research, 55*(8), 2197–2211.

Dhingra, V., Roy, D., & Koster, R. B. M. D. (2015). A cooperative quay crane-based stochastic model to estimate vessel handling time. *Flexible Services & Manufacturing Journal,* 1–28.

Gharehgozli, A. H., Roy, D., & Koster, R. D. (2016). Sea container terminals: New technologies and OR models. *Maritime Economics & Logistics, 18*(2), 103–140.

Grunow, M., Günther, H. O., & Lehmann, M. (2006). Strategies for dispatching AGVs at automated seaport container terminals. *OR Spectrum, 28*(4), 587–610.

Gupta, A., Roy, D., Koster, R. D., & Parhi, S. (2017). Optimal stack layout in a sea container terminal with automated lifting vehicles. *International Journal of Production Research, 55*(13), 3747–3765.

Harik, G. R., Lobo, F. G., & Goldberg, D. E. (2002). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation, 3*(4), 287–297.

Hu, Z. H., Sheu, J. B., & Luo, J. X. (2016). Sequencing twin automated stacking cranes in a block at automated container terminal. *Transportation Research Part C: Emerging Technologies, 69*, 208–227.

Hu, H., Lee, B. K., Huang, Y., Lee, L. H., & Chew, E. P. (2013). Performance analysis on transfer platforms in frame bridge based automated container terminals. *Mathematical Problems in Engineering, 2013*(3), 831–842.

Lau, H. Y. K., & Zhao, Y. (2008). Integrated scheduling of handling equipment at automated container terminals. *Annals of Operations Research, 112*(1), 665–682.

Le, H. M., Yassine, A., & Riadh, M. (2012). Scheduling of lifting vehicle and quay crane in automated port container terminals. *International Journal of Intelligent Information & Database Systems, 6*(6), 516–531.

Li, W., Goh, M., Wu, Y., Petering, M. E. H., Souza, R. D., & Wu, Y. C. (2016). A continuous-time model for multiple yard crane scheduling with last-minute job arrivals. *International Journal of Production Economics, 136*(2), 332–343.

Liang, L., Lu, Z. Q., & Zhou, B. H. (2009). A heuristic algorithm for integrated scheduling problem of container handling system. *International Conference on Computers & Industrial Engineering, IEEE Xplore,* 40–45.

Liang, Y., Lin, L., Gen, M., & Chien, C. F. (2012). A hybrid evolutionary algorithm for FMS optimization with AGV dispatching. *Computers & Industrial Engineering, 296*, 1–14.

Li, Y., Caö, H., & Tan, Y. (2011). Novel method of identifying time series based on network graphs. *Complexity, 17*(1), 13–34.

Liu, C. I., Jula, H., & Ioannou, P. A. (2002). Design, simulation, and evaluation of automated container terminals. *Transactions on Intelligent Transportation Systems, 3*(1), 12–26.

Luo, J., & Wu, Y. (2015). Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals. *Transportation Research Part E Logistics & Transportation Review, 79*, 49–64.

Martín-Soberón, A. M., Monfort, A., Sapiña, R., Monterde, N., & Calduch, D. (2014). Automation in port container terminals. *Procedia - Social and Behavioral Sciences, 160*, 195–204.

Meersmans, P. J. M., & Wagelmans, A. P. M. (2006). Effective algorithms for integrated scheduling of handling equipment at automated container terminals. *Social Science Electronic Publishing, 36*, 1–26.

Meisel, F., & Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review, 45*(1), 196–209.

Meisel, F., & Bierwirth, C. (2013). A framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transportation Science, 47*(2), 131–147.

Mishra, A., & Tripathi, A. K. (2015). Complexity of a problem of energy efficient real-time task scheduling on a multicore processor. *Complexity, 21*(1), 259–267.

Mishra, N., Roy, D., & Ommeren, J. K. V. (2017). A stochastic model for interterminal container transportation. *Transportation Science, 51*(1), 67–87.

Miyamoto, T., & Inoue, K. (2016). Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Computers & Industrial Engineering, 9*, 1–9.

Nguyen, V. D., & Kim, K. H. (2009). A dispatching method for automated lifting vehicles in automated port container terminals. *Computers & Industrial Engineering, 56*(3), 1002–1020.

Nishi, T., Ando, M., & Konishi, M. (2006). Experimental studies on a local rescheduling procedure for dynamic routing of autonomous decentralized AGV systems. *Robotics and Computer-Integrated Manufacturing, 22*(2), 154–165.

Nishi, T., Hiranaka, Y., & Grossmann, I. E. (2011). A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Computers & Operations Research, 38*(5), 876–888.

Pap, E., Bojanić, V., Georgijević, M., & Bojanić, G. (2011). Application of pseudo-analysis in the synchronization of container terminal equipment operation. *Acta Polytechnica Hungarica, 8*(6), 5–21.

Park, T., Choe, R., Ok, S. M., & Ryu, K. R. (2010). Real-time scheduling for twin RMGs in an automated container yard. *OR Spectrum, 32*(3), 593–615.

Pezzella, F., Morganti, G., & Ciaschetti, G. (2008). A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research Systems, 35*(10), 3202–3212.

Qu, H., Xing, K., & Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing, 120*(10), 509–517.

Roy, D., & Koster, R. D. (2015). Stochastic modeling of unloading and loading operations at a container terminal using automated lifting vehicles. *Social Science Electronic Publishing, 5*, 1–47.

Roy, D., Gupta, A., & Koster, R. B. M. D. (2016). A non-linear traffic flow-based queuing model to estimate container terminal throughput with AGVs. *International Journal of Production Research, 54*(2), 1–21.

Sidoti, D., Avvari, G. V., Mishra, M., Zhang, L., Nadella, B. K., Peak, J. E., et al. (2017). A multiobjective path-planning algorithm with time windows for asset routing in a dynamic weather-impacted environment. *IEEE Transactions on Systems Man & Cybernetics Systems, 99*, 1–16.

Skinner, B., Yuan, S., Huang, S., Liu, D., Cai, B., Dissanayake, G., et al. (2013). Optimization for job scheduling at automated container terminals using genetic algorithm. *Computers & Industrial Engineering, 64*(1), 511–523.

Su, L. L., Samuel, M., & Chong, Y. S. (2000). Self-organizing neural network approach for the single AGV routing problem. *European Journal of Operational Research, 121*(1), 124–137.

Vecchia, E. D., Marco, S. D., & Jean-Marie, A. (2012). Illustrated review of convergence conditions of the value iteration algorithm and the rolling horizon procedure for average-cost MDPs. *Annals of Operations Research, 199*(1), 193–214.

Wu, Y., Luo, J., Zhang, D., & Dong, M. (2013). An integrated programming model for storage management and vehicle scheduling at container terminals. *Research in Transportation Economics, 42*(1), 13–27.

Wen, C., Ekşioğlu, S. D., Greenwood, A., & Zhang, S. (2010). Crane scheduling in a shipbuilding environment. *International Journal of Production Economics, 124*(1), 40–50.

Wang, G., Wan, Z., Wang, X., & Lv, Y. (2008). Genetic algorithm based on simplex method for solving linear-quadratic bilevel programming problem. *Computers & Mathematics with Applications, 56*(10), 2550–3255.

Yang, X., Mi, W., Li, X., An, G., Zhao, N., & Mi, C. (2015). A simulation study on the design of a novel automated container terminal. *IEEE Transactions on Intelligent Transportation Systems, 16*(5), 1–11.

Yang, J., Zhang, M., He, B., & Yang, C. (2009). Bi-level programming model and hybrid genetic algorithm for flow interception problem with customer choice. *Computers & Mathematics with Applications, 57*(11–12), 1985–1994.

Zhao, J., & Tang, L. (2011). The simultaneous quay crane and truck scheduling problem in container terminals. In International conference on intelligent computation technology and automation, IEEE, pp. 279–282.

Zhang, J., Ioannou, P. A., & Chassiakos, A. (2006). Automated container transport system between inland port and terminals. *Acm Transactions on Modeling & Computer Simulation, 16*(2), 95–118.

Zhang, H., & Kim, K. H. (2009). Maximizing the number of dual-cycle operations of quay cranes in container terminals. *Computers & Industrial Engineering, 56*(3), 979–992.

Zhang, R., Jin, Z., Ma, Y., & Luan, W. (2015). Optimization for two-stage double-cycle operations in container terminals. *Computers & Industrial Engineering, 83*(C), 316–326.

Zhang, X., Zeng, Q., & Yang, Z. (2016). Modeling the mixed storage strategy for quay crane double cycling in container terminals. *Transportation Research Part E Logistics & Transportation Review, 94*, 171–187.

Yongsheng Yang Pro. Yongsheng Yang, received the Ph.D. degree in systems engineering from Nanjing University of Aeronautics & Astronautics, Nanjing, China, in 1998. He is currently a professor in logistics and engineering College, Shanghai Maritime University, China, the director of Science and Technology Department. Pro. Yang is the candidate of discipline (Mechanical Design and Theory) leader, the associate editor of *Computer-aides Engineering*, the senior member of China Mechanical Engineering Society and the member of IEEE.

Meisu Zhong Meisu Zhong, was born in 1991, she received the master's degree with the College of Economics & Management from the Shanghai Maritime University, Shanghai, China, where she is currently pursuing the Ph.D. degree with the Institute of logistics Science & engineering, is major in the engineering and management of shipping. Her current research interests include the scheduling and controlling of automated container terminal and the control system of the AGV.

Yasser Dessouky Prof. Yasser Dessouky served on the faculty at Miami University, Ohio, prior to joining the faculty at San José State University in 1997. He earned his doctorate and master's degrees in industrial and management systems engineering from Arizona State University and a bachelor's in industrial engineering from the University of Wisconsin-Madison. He also serves as an associate editor for an international journal, Computers and Industrial Engineering, and as an editorial board member on several other

prestigious international journals. He has extensive research and consulting experience in developing models to analyze improvements in the design and efficiency of systems and work flow patterns. He has gained recognition in the areas of simulation modeling and analysis, and quality management, including lean six sigma and integrated supply chain management. His work in supply chain management includes developing models to optimize inventories in the supply network.

Octavian Postolache Instituto de Telecomunicacões/ISCTE-IUL, Lisboa. He graduated in the electrical engineering at the Gh. Asachi Technical University of lasi, Romania, in 1992, and received his Ph.D. degree in 1999 from the same university. In 2000 he became a principal researcher of Instituto de Telecomunicações and an assistant professor of EST/IPS Setubal and ISCTE-IUL Lisboa in 2001 and 2012, respectively. He is the founder and leader of the Pervasive Sensing and Computing research group. His fields of interest are smart sensors, and pervasive sensing and computing. He is author and co-author of 9 patents, 7 books, 14 book chapters, 61 peer reviewed papers in international journals, and more than 200 papers in proceedings of international conferences. He is an ACM professional member, Chair of IEEE MeMeA 2014, Technical Chair of ICST 2014, and an Associate Editor of IEEE Sensors Journal. He received the IEEE I&M outstanding reviewer and IEEE Best Associate Editor Awards.