Transportation Research Part E 46 (2010) 344–353

Contents lists available at ScienceDirect

Transportation Research Part E

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

The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods

Jin Xin Cao [a], Der-Horng Lee [a,*], Jiang Hang Chen [a], Qixin Shi [b]

[a] Department of Civil Engineering, National University of Singapore, Singapore 117576, Singapore
[b] Department of Civil Engineering, Tsinghua University, China 100084, China

ARTICLE INFO

Article history:
Received 19 February 2009
Received in revised form 17 June 2009
Accepted 18 August 2009

Keywords:
Container terminal operations
Benders' decomposition
Scheduling
Combinatorial optimization
Mixed-integer programming

ABSTRACT

This paper proposes a novel integrated model for yard truck and yard crane scheduling problems for loading operations in container terminal. The problem is formulated as a mixed-integer programming model. Due to the computational intractability, two efficient solution methods, based on Benders' decomposition, are developed for problem solution; namely, the general Benders' cut-based method and the combinatorial Benders' cut-based method. Computational experiments are conducted to evaluate the effectiveness of the proposed solution methods.

1. Introduction

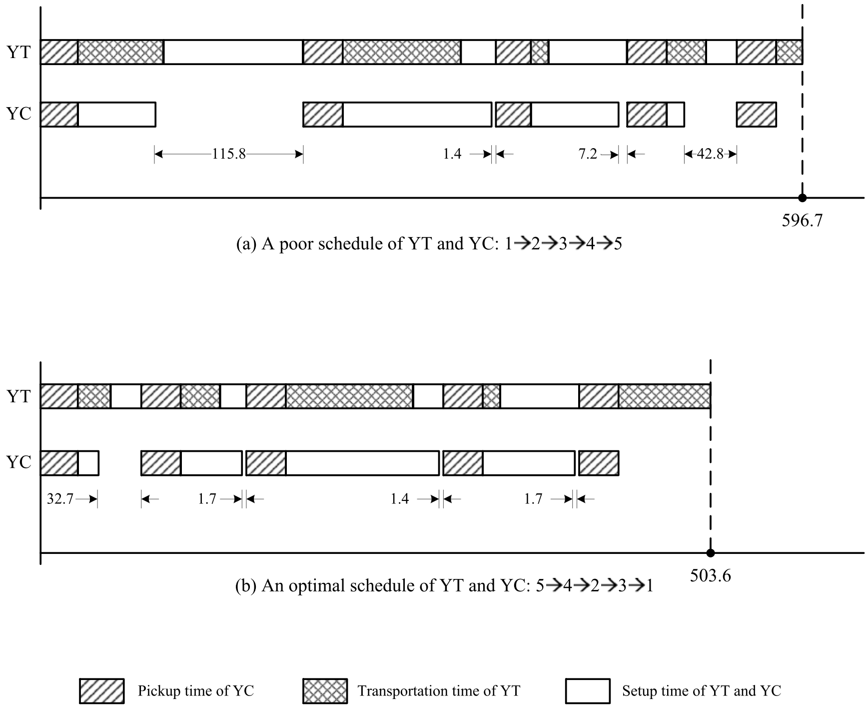
Currently, container terminals are facing new challenges brought about by the development of maritime transportation. On the one hand, to reduce the transportation cost, mega-vessels capable of carrying 10,000–12,000 TEUs (20-foot equivalent units) and beyond have been deployed increasingly by shipping companies. The handling of mega-vessels introduces whole new challenges in container terminal operations. For example, the efficiency of stacking and moving large number of containers in terminal area is an immediate concern in container terminal operations. On the other hand, new equipments, such as double-trolley quay cranes, automated stacking cranes (ASC), and automated guided vehicles (AGV), have been utilized by container terminals to improve operating efficiency and reduce operating costs.

This boosts the studies on new operational approaches that can help exert the productivity of these equipments and hence improve the efficiency of container terminals. Research on how to maximize the utilities of scarce resources in container terminals, including equipment, land and time, has been studied for decades. Previous research decomposed the terminal operations into subproblems for the convenience of modeling and solution. Generally, those subproblems include berth allocation, quay crane (QC) scheduling, storage allocation, yard crane (YC) scheduling and yard truck (YT) scheduling. This paper dedicates the efforts in addressing YT scheduling and YC scheduling problems.

The YT scheduling and YC scheduling are two highly interrelated decision problems faced by yard operations in container terminals. YT plays a role of interface between quay side and yard side operations. For loading operations of outbound containers, the makespan highly depends on the synchronization of YT scheduling and YC scheduling. A simple example with one YT and one YC loading five containers and the corresponding values of parameter are illustrated in Fig. 1. Fig. 1a shows an arbitrary schedule of YT and YC without synchronization, and the working sequence of jobs by the YT and YC is

* Corresponding author.
E-mail address: dhl@nus.edu.sg (D.-H. Lee).

1366-5545/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2009.08.012



s_{ij}^1

j \ i	1	2	3	4	5
1	$+\infty$	109.4	19.1	70.6	80.5
2	69.1	$+\infty$	24.1	58.4	67.1
3	59.1	100.1	$+\infty$	61.2	71.1
4	35.2	18.3	85.1	$+\infty$	23.2
5	45.5	35.4	64.0	23.3	$+\infty$

s_{ij}^2

j \ i	1	2	3	4	5
1	$+\infty$	62.5	70.7	28.0	32.6
2	62.5	$+\infty$	115.4	48.6	36.3
3	70.7	115.4	$+\infty$	67.4	79.7
4	28.0	48.6	67.4	$+\infty$	12.4
5	32.6	36.3	79.7	12.4	$+\infty$

i	1	2	3	4	5
t_i	68.9	92.7	13.4	32.0	21.9

Fig. 1. Illustration of YT and YC schedules.

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$; while Fig. 1b is an optimal schedule of the integrated YT and YC scheduling problem for the same example. By solving the mathematical program, the optimal working sequence of jobs by the YT and YC is $5 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 1$. From Fig. 1, it can be observed that the makespan can be reduced by 16%, and the total waiting time of YC is reduced from 167.2 to 37.5. Therefore, by integrating YT and YC scheduling into a whole, the unproductive time of YT and YC can be greatly reduced, so that the productivity of container terminal is therefore improved.

This paper proposes an *integrated YT and YC scheduling problem* (*i*-YTYCSP), which aims to minimize the makespan of loading outbound containers by synchronizing the operations of YTs and YCs. This introductive section is followed by the review of related literatures. The *i*-YTYCSP is formulated as a *mixed-integer programming* (MIP) model in Section 3, in which the computational complexity of the *i*-YTYCSP is investigated as well. The *general Benders' cut-based* (GBC-based) solution method is developed in Section 4. A more effective algorithm, namely the *combinatorial Benders' cut-based* (CBC-based) solution method is proposed in Section 5. Computational experiments are conducted in Section 6 to evaluate the effectiveness of the proposed solution methods, followed by Section 7 to conclude this paper.

2. Literature review

Issues unique to container terminal operations are receiving more and more attentions due to the increasing importance of marine transportation system. One of the most important objectives, which is frequently considered in literatures, is to increase the throughput of terminals or, in particular, to decrease the turnaround time of container vessels (Steenken et al., 2004). Effectively allocating scarce resources (such as berths and storage locations) and scheduling vehicles and handling equipments directly determine the efficiency of a container terminal.

Scheduling vehicles and equipments, such as YTs, QCs and YCs, frequently arises in logistic systems and these problems have been extensively studied under different settings (Bramel and Simchi-Levi, 1997). Scheduling problem faced in daily operations of container terminals can be reduced to two well-studied problems, namely the *vehicle routing problems* (VRP) and the *machine scheduling problems* (MSP). Unfortunately, the results reported in literatures related to VRP and MSP are not directly applicable to container terminals due to its fundamental differences.

In the field of scheduling transporters in container terminals, Kim and Bae (2004) studied the dispatching of AGVs by utilizing information of location and time of future delivery tasks. The static version of the problem was formulated as a one-to-one assignment problem with mixed-integer programming model. A heuristic algorithm was developed to solve the proposed problem.

Ng et al. (2007) addressed the problem of scheduling a set of YTs at a container terminal in order to minimize the makespan. YTs had to perform a set of transportation jobs with sequence-dependent processing times and different ready times. The problem was solved by a Genetic Algorithm (GA).

Bish et al. (2001) proposed a vehicle-scheduling–location problem, in which the crane job sequence was predetermined and generated precedence constraints for discharging of containers.

Bish (2003) extended Bish et al. (2001) by considering multiple QCs and a pool of YTs. In the multiple-crane-constrained scheduling problem, both loading and unloading processes were considered.

Lee et al. (2008) proposed an integrated model for YT scheduling and storage allocation model for import containers. The problem was formulated as an MIP that considered the two problems into a whole. The objective was to minimize the makespan of operations. Due to the $\mathcal{NP}$-hardness of the problem, a GA and a dedicated heuristic algorithm were developed to solve the problem.

Another significant subproblem in container terminal is the YC scheduling problem. Kim and Kim (1999a) studied the problem of routing a single YC. An MIP was proposed to determine the number of containers to be picked up in each slot as well as the sequence of the locations to be visited by the straddle carrier. Kim and Kim (1999b) extended the approach proposed by Kim and Kim (1999a) for the application of straddle carrier, instead of YC.

Ng and Mak (2005) studied the YC scheduling problem to perform a given set of loading/unloading jobs with different ready times, so as to minimize the sum of job waiting times. A *branch-and-bound* (B&B) algorithm was proposed to achieve the exact solution to the scheduling problem.

Ng (2005) proposed an MIP model for the YC scheduling problem with the consideration of inter-crane interference. A dynamic programming-based heuristic was developed to solve the scheduling problem.

Lee et al. (2007) solved the problem of scheduling two YCs for loading export containers with the simulated annealing (SA) approach. The two gantry cranes served the loading operations of one QC. The schedule determined the container bay visiting sequences as well as the number of containers picked up simultaneously.

Cao et al. (2008) focused on providing an efficient operation strategy for the double-rail-mounted gantry systems to load outbound containers. An MIP model was developed to formulate the problem. A greedy heuristic algorithm, an SA algorithm and a combined double-rail-mounted gantry scheduling heuristic were designed to solve the proposed problem.

Most of previous studies focus on one of the decision problems in container terminal operations, due to the computational intractability. Admittedly, synchronization of different material handling systems, *i.e.*, QCs, YCs and YTs, is crucial to the efficiency of container terminals. Researchers start to investigating the integrated scheduling approach for container terminal operations, with which a system composes of two or more subsystems, instead of one, are optimized (Stahlbock and Voß, 2008). However, to our best knowledge, no effort has been dedicated to the *i*-YTYCSP, as we attempt in this paper. Since during the loading process, YTs and YCs operations are to serve the QCs, so that all containers can be loaded by the QCs according to their loading plans. Thus, YCs and YTs are two interrelated subsystems; the compatibility and synchronization of which will enhance the efficiency of the entire container terminal system.

3. Problem description and formulation

In this section, we formulate the integrated YT and YC scheduling problem (*i*-YTYCSP) as a mixed-integer programming (MIP) model. In this paper, we refer the operation of each container as a *job*. The following assumptions are imposed in the *i*-YTYCSP formulated:

1. Only loading operations of outbound containers are considered.
2. The storage location of each container (the origin of the job) is given.
3. The locations of QCs by which containers are loaded (the destinations of jobs) are given.
4. Each container requires different pickup time by YCs, and every container can be accessed by any YC with the same amount of time.
5. After completion of the current job, both YC and YT move from the destination of the current job to the origin of their succeeding jobs.
6. The travel speed of YCs is different from the travel speed of YTs.
7. The capacity of YTs is equal to 1, which means each YT can move one job at a time.
8. No interference of YTs is considered.
9. The containers can be handled by QCs in any order, and due dates of containers are not considered.
10. The containers are available from the yard in any order within pre-determined time.
11. The multiple YCs do not interfere with each other.

Let $\mathcal{J}$ be the set of jobs with cardinality of $|\mathcal{J}| = n$. There are k YTs and m YCs deployed to load the outbound containers onto the vessels. Let $\mathcal{L}$ be the set of all locations (origins and destinations of containers). Additionally, we define two dummy jobs, indexed by 0 and $n+1$, to represent the initial and final states of YTs and YCs. The makespan of all jobs in $\mathcal{J}$ is therefore equal to the completion time of Job $n+1$. We define three new sets by extending $\mathcal{J}$:

$$\mathcal{J}_1 = \mathcal{J} \cup \{0\};$$
$$\mathcal{J}_2 = \mathcal{J} \cup \{n+1\};$$
$$\mathcal{J}_3 = \mathcal{J}_1 \cup \mathcal{J}_2.$$

In the mathematical model the following indices are used:

i,j,h The indices of jobs, *i.e.*, $i,j,h \in \mathscr{J}_3$
u, v The indices of locations (origins and destinations), *i.e.*, $u, v \in \mathscr{L}$

The following parameters are used in the model:

p_i the pickup time of Job i by a YC
δ_{uv} the distance between Location u and Location v
o_i the origin of Job i, and $o_i \in \mathscr{L}$
d_i the destination of Job i, and $d_i \in \mathscr{L}$
v_1 the travel speed of YTs
v_2 the travel speed of YCs
s^1_{ij} the setup time of YTs from Job i to Job j
s^2_{ij} the setup time of YCs from Job i to Job j
t_i the transportation time of Job i by YTs
τ the processing time of QCs, *i.e.*, the time required to transfer a container from a YT to a QC

The setup time of YTs from Job i to Job j, s^1_{ij}, is defined as the travel time of YTs from the destination of Job i, d_i, to the origin of Job j, o_j, as

$$s^1_{ij} = \frac{\delta_{d_i,o_j}}{v_1}. \tag{1}$$

The setup time of YCs from Job i to Job j, s^2_{ij}, is defined as the travel time of YCs from the origin of Job i, d_i, to the origin of Job j, o_j, as

$$s^2_{ij} = \frac{\delta_{o_i,o_j}}{v_2}. \tag{2}$$

The transportation time of Job i, t_i, is defined as the travel time of YTs from the origin of Job i, o_i, to the destination, d_i, as Eq. (3):

$$t_i = \frac{\delta_{o_i,d_i}}{v_1}. \tag{3}$$

The pickup time of Job i by a YC, denoted by p_i, is the time duration that a YC needs to pick up the container from its storage location in yard and load it onto the YT. The pickup time of Job i by a YC also includes the retrieval time required by the YC to access Job i.

Decision variables in the model include:

$$\begin{aligned} x_{ij} &= 1, \quad \text{if Job } j \text{ is moved immediately after Job } i \text{ by the same YT;} \\ &= 0, \quad \text{otherwise}, \quad \forall i \in \mathscr{J}_1 \text{ and } j \in \mathscr{J}_2; \\ y_{ij} &= 1, \quad \text{if Job} j \text{ is picked up immediately after Job } i \text{ by the same YC;} \\ &= 0, \quad \text{otherwise}, \forall i \in \mathscr{J}_1 \text{ and } j \in \mathscr{J}_2; \\ c_i &= \text{The completion time of Job } i, \quad \forall i \in \mathscr{J}_3. \end{aligned}$$

If a YT performs Job h immediately after it completes Job i, (*i.e.*, $x_{ih} = 1$), and a YC performs Job h immediately after it completes Job j, (*i.e.*, $y_{jh} = 1$), we have the following constraints to the completion time of Job h:

$$c_h = \max\left\{c_i + s^1_{ih} + p_h + t_h + \tau, c_j - t_j + p_h + t_h + s^2_{jh}\right\}. \tag{4}$$

The i-YTYCSP can be formulated as the following MIP:

$$[\text{P1}]\ \min \quad c_{n+1} \tag{5}$$

$$\text{s.t.} \quad \sum_{j \in \mathscr{J}_2} x_{ij} = 1, \quad \forall i \in \mathscr{J}, \tag{6}$$

$$\sum_{i \in \mathscr{J}_1} x_{ij} = 1, \quad \forall j \in \mathscr{J}, \tag{7}$$

$$\sum_{j \in \mathscr{J}} x_{0j} = k, \tag{8}$$

$$\sum_{i \in \mathscr{J}} x_{i,n+1} = k, \tag{9}$$

$$\sum_{j \in \mathscr{J}_2} y_{ij} = 1, \quad \forall i \in \mathscr{J}, \tag{10}$$

$$\sum_{i \in \mathscr{J}_1} y_{ij} = 1, \quad \forall j \in \mathscr{J}, \tag{11}$$

$$\sum_{j \in \mathscr{J}} y_{0j} = m, \tag{12}$$

$$\sum_{i \in \mathscr{J}} y_{i,n+1} = m, \tag{13}$$

$$(c_j - c_i) + M(1 - x_{ij}) \geqslant \alpha_{ij}, \quad \forall i \in \mathscr{J}_1 \text{ and } j \in \mathscr{J}_2, \tag{14}$$

$$(c_j - c_i) + M(1 - y_{ij}) \geqslant \beta_{ij}, \quad \forall i \in \mathscr{J}_1 \text{ and } j \in \mathscr{J}_2, \tag{15}$$

$$x_{ij}, y_{ij} \in \{0, 1\}, \quad \forall i \in \mathscr{J}_1 \text{ and } j \in \mathscr{J}_2, \tag{16}$$

$$c_i \in \Re^+, \quad \forall i \in \mathscr{J}_3. \tag{17}$$

The objective is to minimize the completion time of Job $n+1$, which is equal to the makespan of all jobs. Constraints (6) and (7) mean for each Job $i \in \mathscr{J}$, there are exactly one preceding job and one succeeding job that are assigned to the same YT as Job i. Constraints (8) and (9) guarantee that there are exactly k YTs are deployed. Constraints (10) and (11) represent for each Job $i \in \mathscr{J}$, there are exactly one preceding job and one succeeding job that are assigned to the same YC as Job i. Constraints (12) and (13) guarantee that there are exactly m YTs are deployed. Eq. (4) is translated into the Constraints (14) and (15), namely the *scheduling constraints*, where

$$\alpha_{ij} = \begin{cases} p_j + t_j + \tau, & \text{for } i = 0, j \in \mathscr{J}; \\ p_j + t_j + \tau + s^1_{ij}, & \text{for } i, j \in \mathscr{J}; \\ 0, & \text{for } i \in \mathscr{J}, j = n+1; \\ \infty, & \text{for } i = 0, j = n+1; \end{cases} \tag{18}$$

$$\beta_{ij} = \begin{cases} p_j + t_j + \tau, & \text{for } i = 0, j \in \mathscr{J}; \\ p_j + t_j - t_i + s^2_{ij}, & \text{for } i, j \in \mathscr{J}; \\ 0, & \text{for } i \in \mathscr{J}, j = n+1; \\ \infty, & \text{for } i = 0, j = n+1. \end{cases} \tag{19}$$

We also have $c_0 = 0$, *i.e.* all YTs and YCs are available at the beginning of the planning horizon. To simplify, we call Constraints (6)–(9) the *connectivity constraints for YTs*, and Constraints (10)–(13) the *connectivity constraints for YCs*. Constraints (16) and (17) state the restrictions of variables.

Actually, the solution of [P1] can be represented by a network, as can be seen in Fig. 2. Each node (excluding 0 and 11) has exactly two inflow arcs (one for YT routes and one for YC routes) and two outflow arcs (one for YT routes and one for YC routes). For example the completion time of Job 7 depends on the completion time of the two jobs corresponding to its inflow arcs, i.e. Job 1 and Job 6. Since we assume that the YTs and YCs are identical, respectively, it is not necessary to distinguish YTs and YCs in the model formulation. After establishing the one-to-one assignment relationship, the YT and YC serving a container can be clearly determined.

From the view point of scheduling theory, the proposed *i*-YTYCSP is a two-stage flexible flowshop problem with the objective of minimizing the makespan and a series of side constraints. The two-stage flexible flowshop problem has already been

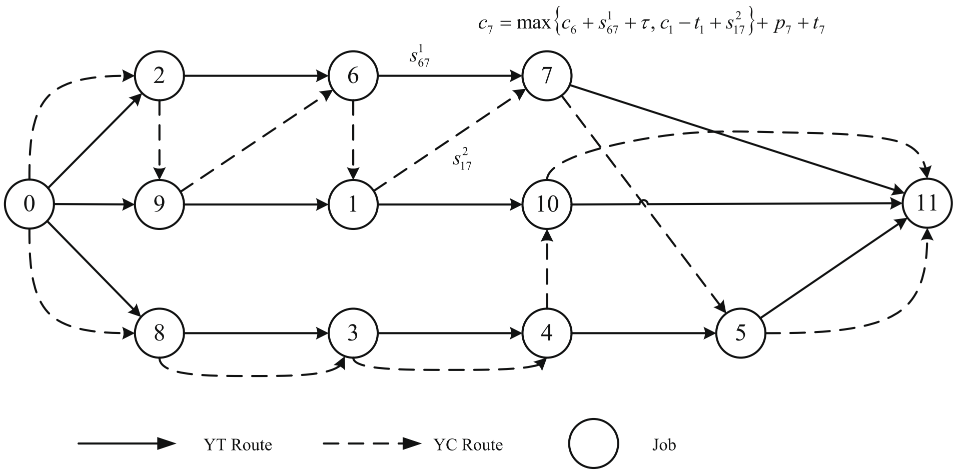


Fig. 2. A network representation of a solution to [P1].

proved to be $\mathcal{NP}$-hard in the strong sense, even in the case where there are only two machines in one stage and there is only a single machine in the other stage (Hoogeveen et al., 1996). Therefore, the proposed *i*-YTYCSP is $\mathcal{NP}$-hard in the strong sense.

Due to the computational intractability, it is unlikely that exact optimal solutions to the *i*-YTYCSP can be obtained in polynomial time using general solution algorithms, such as B&B and cutting plane algorithms. Thus, in the following sections, we propose two solution methods for the *i*-YTYCSP, namely the general Benders' cut-based method and the combinatorial Benders' cut-based method, both of which are proven to be much more efficient than B&B in CPLEX.

4. General Benders' cut-based method

Benders' decomposition is a partitioning procedure for solving the MIP that was developed by Benders (1962). Benders' decomposition has been successfully applied to large-scale systems and stochastic programming (Cordeau et al., 2001; Geoffrion and Graves, 1974). To be concise, vector notations are defined as followings:

x the vector of $\{x_{ij}\}_{i\in\mathcal{I}_1, j\in\mathcal{I}_2}$
y the vector of $\{y_{ij}\}_{i\in\mathcal{I}_1, j\in\mathcal{I}_2}$
c the vector of $\{c_i\}_{i\in\mathcal{I}_3}$

At first, we reformulate [P1] as a master problem namely [Master].

$$[\text{Master}]\ \min \quad \mathbf{0x} + \mathbf{0y} + z(\mathbf{x},\mathbf{y}), \qquad (20)$$
$$\text{s.t.} \quad \text{Constraints (6)–(13) and (16)},$$

where $z(\mathbf{x},\mathbf{y})$ is a function of **x** and **y**, and can be obtained by solving the subproblem, denoted by [P-Sub].

$$[\text{P-Sub}] z_P(\mathbf{x},\mathbf{y}) = \min \quad c_{n+1}, \qquad (21)$$
$$\text{s.t.} \quad \text{Constraints (14) and (15) and (17)}.$$

The dual subproblem can be formulated as

$$[\text{D-Sub}] z_D(\mathbf{x},\mathbf{y}) = \max \quad \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij} + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}, \qquad (22)$$

$$\text{s.t.} \quad \sum_{j=0}^{n}(\mu_{ji} + \pi_{ji}) - \sum_{j=1}^{n+1}(\mu_{ij} + \pi_{ij}) \leqslant 0, \quad \forall i = 1,\ldots,n, \qquad (23)$$

$$\sum_{i=0}^{n}(\mu_{i,n+1} + \pi_{i,n+1}) \leqslant 1, \qquad (24)$$

$$\mu_{ij}, \pi_{ij} \in \Re^+, \quad \forall i \in \mathcal{I}_1 \text{ and } j \in \mathcal{I}_2. \qquad (25)$$

The [P-Sub] is parameterized by **x** and **y**. Assuming [P1] is finite implies that [P-Sub] is finite for at least one feasible solution of [Master]. Applying the duality theorem, [D-Sub] has to be feasible. Assuming [P1] is feasible implies that [P-Sub] is feasible for at least one feasible solution of [Master]. Applying the duality theorem again, [D-Sub] has to be finite.

Let $\boldsymbol{\mu}^\iota, \boldsymbol{\pi}^\iota$ $(\iota = 1,\ldots,p)$ be the extreme points of [D-Sub], and $\boldsymbol{\mu}^\iota, \boldsymbol{\pi}^\iota$ $(\iota = p+1,\ldots,p+q)$ be the extreme rays of the dual feasible region.

Dual is finite if and only if

$$\sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^\iota + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^\iota \leqslant 0, \quad \forall(\iota = p+1,\ldots,p+q). \qquad (26)$$

We append these constraints to [Master] to ensure that the [D-Sub] is always bounded ([P-Sub] always feasible).

[D-Sub] can now be written as

$$z(\mathbf{x},\mathbf{y}) = \max_{\iota=1,\ldots,p} \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^\iota + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^\iota. \qquad (27)$$

We outer-linearize this function:

$$z^\iota(\mathbf{x},\mathbf{y}) = \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^\iota + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^\iota, \quad \forall(\iota = 1,\ldots,p), \qquad (28)$$

$$z^\iota(\mathbf{x},\mathbf{y}) \leqslant z(\mathbf{x},\mathbf{y}), \quad \forall(\iota = 1,\ldots,p). \qquad (29)$$

Then the full master problem can be formulated as

$$[\text{FMP}]\ \min \quad \theta, \tag{30}$$

$$\text{s.t.} \quad \text{Constraints}(6)-(13)\text{and}(16),$$

$$\theta \geqslant \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^{\imath} + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^{\imath}, \quad \forall(\imath = 1,\ldots,p), \tag{31}$$

$$0 \geqslant \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^{\imath} + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^{\imath}, \quad \forall(\imath = p+1,\ldots,p+q), \tag{32}$$

$$\theta \in \Re^{+} \tag{33}$$

We call Constraints (31) and (32) the *general Benders' cuts* (GBCs). The *i*-YTYCSP can be solved by iteratively generating GBCs. The following shows the detailed steps of the GBC-based method:

Step 1. At Iteration $\imath$, the partial [FMP] (with only $\imath - 1$ GBCs) is solved, and obtain $\hat{\mathbf{x}}^{\imath}, \hat{\mathbf{y}}^{\imath}$ and $\theta^{\imath}$.
Step 2. With $\hat{\mathbf{x}}^{\imath}, \hat{\mathbf{y}}^{\imath}$ solve [D-Sub].

If [D-Sub] is unbounded, obtain the extreme ray $\mu^{\imath}$ and $\pi^{\imath}$ and generate a GBC to the partial [FMP], as Eq. (34). Let $k = k+1$ and go to *Step 1*.

$$0 \geqslant \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^{\imath} + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^{\imath} \tag{34}$$

If [D-Sub] has an optimal solution, go to *Step 3*.

Step 3. The lower bound, denoted by $Z_{LB}^{\imath}$, can be calculated by:

$$Z_{LB}^{\imath} = \theta^{\imath}, \tag{35}$$

and the upper bound, denoted by $Z_{UB}^{\imath}$, can be calculated by:

$$Z_{UB}^{\imath} = z_D(\boldsymbol{x}^{\imath}, \boldsymbol{y}^{\imath}). \tag{36}$$

If the *GAP*, calculated by:

$$GAP(\%) = \frac{Z_{UB}^{\imath} - Z_{LB}^{\imath}}{Z_{UB}^{\imath}} \times 100, \tag{37}$$

is less than the tolerance, denoted by *TOL*, the optimal solution is found. Go to *Step 4*; else, obtain the extreme point $\mu^{\imath}$ and $\pi^{\imath}$ and generate a GBC to the partial [FMP], as Eq. (38). Let $k = k+1$ and go to *Step 1*.

$$\theta \geqslant \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\alpha_{ij} - M(1-x_{ij})]\mu_{ij}^{\imath} + \sum_{i=0}^{n}\sum_{j=1}^{n+1}[\beta_{ij} - M(1-y_{ij})]\pi_{ij}^{\imath} \tag{38}$$

Step 4. Stop.

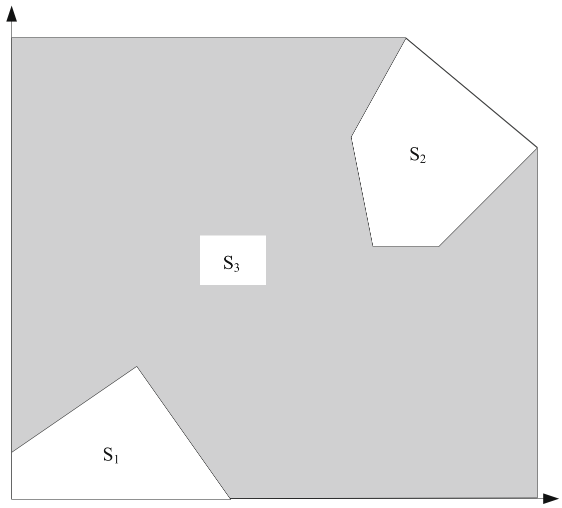


Fig. 3. Illustration of the big-*M* convex hull.

Following the above mentioned method, the i-YTYCSP can be solved iteratively. As a matter of fact, the GBC-based method can be viewed as a tool to accelerate the solution of the *linear programming* (LP) relaxation, but not to improve its quality. In [P1], Constraints (14) and (15) use the big-M method to represent the logic relationship between indicator variables $\mathbf{x}$, $\mathbf{y}$ and the continuous variable $\mathbf{c}$. As can be seen from Fig. 3, Space S_1 is the convex hull for indicator variables having the value of 1, and Space S_2 is the convex hull for indicator variables having the value of 0. The actual feasible region is $S_1 \cup S_2$. However, by applying the big-M formulation, the convex hull becomes S_3 (in gray), which highly weakens the LP relaxation of the problem.

Though, by using the GBC-based method, we can get rid of the continuous variables, but the resulting cuts are not strong enough and still depends on the big-M value. Therefore, in the next section, we will develop the combinatorial Benders' cuts to strengthen the cuts generated by the subproblem. The combinatorial Benders' cuts no longer depend on the big-M value, thus can be much more efficient than GBCs.

5. Combinatorial Benders' cut-based method

Combinatorial Benders' cut (CBC) is proposed by Hooker (2000), which derives Benders' cuts from *minimal set of inconsistencies*. Codato and Fischetti (2006) extended Hooker (2000) so as to solve the MIP with special structures. In this section, we develop the CBC-based method to strengthen our cut generated by the subproblem. The following notations will be used in the development of the CBC-based method for the i-YCYTSP:

UB the value of the incumbent solution
ϵ a sufficiently small positive value

In CBC algorithm, the master problem is the same as GBC, *i.e.*, [Master]. Instead of examining the subsystem, in CBC algorithm, the following linear system is introduced, which is parameterized by $\mathbf{x}$ and $\mathbf{y}$:

$$Sub(\mathbf{x},\mathbf{y}) := \begin{cases} \text{Eqs. (14), (15) and (17)} \\ c_{n+1} \leqslant UB - \epsilon \end{cases} \tag{39}$$

Let $\mathscr{C}$ be the *minimal infeasible subsystem* (MIS) of $Sub(\mathbf{x},\mathbf{y})$, *i.e.*, any inclusion-minimal set of row indices, *i.e.*, (i,j), of system $Sub(\mathbf{x},\mathbf{y})$ such that linear system has no feasible (continuous) solution $\mathbf{c}$.

At the iterations where $Sub(\mathbf{x}^*,\mathbf{y}^*)$ is feasible, we can find an optimal solution $\mathbf{c}^*$ of the LP problem [P-Sub] and update the best incumbent solution by $(\mathbf{x}^*,\mathbf{y}^*,\mathbf{c}^*)$.

If the linear system is infeasible, instead, $(\mathbf{x}^*,\mathbf{y}^*)$ itself is infeasible for program [P1]. We therefore look for an MIS of $Sub(\mathbf{x}^*,\mathbf{y}^*)$, indexed by $\mathscr{C}$, and observe that at least one binary variables x_{ij}, y_{ij} $(\forall (i,j) \in \mathscr{C})$ has to be changed to break the infeasibility. This condition can be translated by the following linear inequality in the ($\mathbf{x}$,$\mathbf{y}$) space, that we call the CBC:

$$\sum_{(i,j)\in\mathscr{C}_1:x^*_{ij}=0} x_{ij} + \sum_{(i,j)\in\mathscr{C}_2:y^*_{ij}=0} y_{ij} + \sum_{(i,j)\in\mathscr{C}_1:x^*_{ij}=1} (1-x_{ij}) + \sum_{(i,j)\in\mathscr{C}_2:y^*_{ij}=1} (1-y_{ij}) \geqslant 1, \tag{40}$$

where $\mathscr{C} = \mathscr{C}_1 \cup \mathscr{C}_2$. The subset $\mathscr{C}_1$ contains indices of rows corresponding to Constraint (14); and the subset $\mathscr{C}_2$ contains indices of rows corresponding to Constraint (15).

The overall method will stop when the current master problem, which looks for a feasible ($\mathbf{x}$,$\mathbf{y}$) to improve the incumbent, becomes infeasible. Iterating the procedure produces an exact solution method based on Benders' decomposition.

6. Computational experiments

To evaluate the effectiveness of GBC-based method and CBC-based method, we randomly generate 20 computational examples based on the layout of Keppel Terminal in Singapore. The origins of containers are randomly selected in yard blocks, and the destinations of containers are randomly determined along the quay side. It is assumed that the distance between any two locations is the distance along the shortest path. We assume that the travel speed of YTs is 5 m/s and the travel speed of YCs is 4 m/s. The pickup time of containers is assumed to follow the uniform distribution, ranging from 90 s to 120 s (Stahlbock and Voß, 2008). We assume the processing time of QC is 60 s. Table 1 shows the details of computational examples.

We implement our methods in C++ and embedded it within the ILOG Concert Technology frame work, based on ILOG CPLEX 9.1.

We solve Examples 1–20 by three different methods:

1. Solve [P1] directly with the B&B algorithm with CPLEX.
2. Solve the i-YTYCSP with the GBC-based method.
3. Solve the i-YTYCSP with the CBC-based method.

Table 1
Details of computational examples.

Example	No. Jobs	No. YTs	No. YCs
1	10	2	2
2	10	3	2
3	10	4	3
4	20	3	2
5	20	4	3
6	20	5	4
7	30	4	3
8	30	5	4
9	30	6	5
10	50	4	3
11	50	5	4
12	100	8	4
13	100	10	5
14	100	20	10
15	200	10	10
16	200	20	15
17	200	40	30
18	500	20	10
19	500	40	20
20	500	60	40

Table 2
Problems solved to optimality by all methods.

Example	Opt. (s)	CPLEX (h:m:s)	GBC (h:m:s)	CBC (h:m:s)
1	1445	18:24:54	10:03:07	00:05:45
2	1138	20:07:14	09:05:20	00:07:01
3	1041	18:15:24	09:12:15	00:08:12
4	2015	30:15:01	20:18:30	00:10:23
5	1975	28:01:29	18:14:12	00:12:13
6	1543	31:45:10	21:16:20	00:13:35
7	2146	32:34:30	25:30:20	00:15:01
8	1953	34:01:33	29:19:30	00:17:08
9	1746	35:03:25	30:24:05	00:21:09

Table 3
Problems solved to optimality by CBC but not by CPLEX and GBC.

Example	CPLEX			GBC			CBC	
	Time (h:m:s)	Best solution (s)	Gap (%)	Time (h:m:s)	Best solution (s)	Gap (%)	Opt. (s)	Time (h:m:s)
10	36:04:12	3206	23	36:00:27	3411	11	3018	00:22:00
10	36:04:12	3206	23	36:00:27	3411	13	3018	00:22:00
11	36:05:15	3037	21	36:01:31	3264	12	2967	00:25:00
12	36:01:32	5352	43	36:03:31	5939	18	5078	00:30:01
13	36:05:01	4720	32	36:02:32	5290	17	4533	00:32:34
14	36:08:23	4397	37	36:01:22	4724	19	4038	00:21:35
15	36:03:31	5288	55	36:05:23	5091	9	4675	00:35:21
16	36:02:45	4913	26	36:02:34	4938	12	4354	00:33:42
17	36:01:03	4125	40	36:01:33	4218	15	3679	00:32:56
18	36:03:34	11,204	18	36:04:31	12,430	19	10,469	00:41:21
19	36:07:57	10,605	25	36:03:23	10,549	18	8943	00:40:33
20	36:00:32	8812	42	36:05:31	8760	13	7803	00:45:30

All experiments are performed on a PC Intel Core 2 2.40 GHz with 3 GByte RAM and Windows XP operating system. A time limit of 36 CPU hours was imposed for each run.

Only Examples 1–9 can be solved to optimality by CPLEX and GBC-based method. The computational results of Examples 1–9 are shown in Table 2. The second column shows the optimal value of each example. Columns 3–5 depict the computational time of each example by CPLEX, GBC-based method and CBC-based method, respectively. It can be observed that, for Examples 1–9, the computational time of CPLEX is 1.57 times of the computational time of GBC-based method and 141.35 times of the computational time of CBC-based method on average. The maximum computational time of CPLEX is more than 35 h for 30 containers, which is far from the practical requirement in container terminal operations. Though GBC-based method is more efficient than CPLEX, it is still incapable of handling problem with practical scale. In contrast, the CBC-based method is very effective so as to obtain the optimal value of 30 containers within half an hour.

CPLEX and GBC-based methods cannot solve Examples 10–20 to optimality, but CBC-based method still can obtain the optimal solution within 36 h. Table 3 shows the computational results for Examples 10–20. Column 2 is the computational time by CPLEX for each example, and Column 3 is the best solution obtained by CPLEX within 1 h. Column 4 computes the Gap between the best solution and the best bound by CPLEX. Columns 5 and 6 represent the computational time and the best solution by the GBC-based method for each example. Column 7 shows the Gap of each example by GBC-based, which is calculated by Eq. (40). Column 8 shows the optimal value obtained by CBC-based method and Column 10 shows the computational time of CBC-based method for each example. It can be observed from Table 3 that the average Gap by CPLEX is 33% and the average Gap by GBC-based method is 14.8%, which is approximately half of the Gap by CPLEX. For all examples, the CBC-based method can obtain the optimal solution, and the longest computational time is about 45 min. Therefore, the CBC-based method is considered as efficient for daily operations.

7. Conclusions

This paper has proposed an integrated yard truck and yard crane scheduling problem (*i*-YTYCSP). In the *i*-YTYCSP, the objective is to minimize the makespan of loading all outbound containers in the planning horizon. The proposed *i*-YTYCSP is $\mathcal{NP}$-Hard, which means it is unlikely that the problem can be efficiently solved with exact solution algorithms. For this reason, two different Benders' decomposition based methods are developed for the *i*-YTYCSP, namely the general Benders' cut-based (GBC-based) method and the combinatorial Benders' cut-based (CBC-based) method. In average, the computational time of CPLEX method is about 1.57 times longer than the time required when the GBC-based method is used to solve the problem. The CBC-based method can highly improve the LP relaxation of the master problem by eliminating the big-M constraints. Thus, the CBC-based method is deemed to be efficient that practical scale problems can be handled with reasonable computational time.

The *i*-YTYCSP proposed in this paper provides a novel idea that can improve the efficiency of container terminal operations. Container terminal operations have been studied for decades, and now most of the researchers have noticed the importance and value of the integrated modeling approach. Benders' decomposition has been successfully applied in solving multi-stage optimization problems, to which most of integrated models in container terminal operations can be applied. Therefore, the methods proposed in this paper based on Benders' decomposition have the potential to handle a series of integrated optimization problems in container terminal operations. How to apply Benders' decomposition based method on other integrated models is a valuable future research topic. In addition, investigating the feasibility of other solution methods, such as column generation, Lagrangian relaxation and constraint satisfaction, in solving integrated models in container terminal operations is another interesting topic. Various complicated and practical constraints, such as the due date of containers, retrieval time containers, interferences of YTs and YCs should be considered by extending the methods proposed in this paper for the future research.

References

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4, 238–252.

Bish, E.K., 2003. A multiple-crane-constrained scheduling problem in a container terminal. European Journal of Operational Research 144, 83–107.

Bish, E.K., Leong, T.-Y., Li, C.-L., Ng, W.C., Simchi-Levi, D., 2001. Analysis of a new vehicle scheduling and location problem. Naval Research Logistics 48, 363–385.

Bramel, J., Simchi-Levi, D., 1997. The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management. Springer-Verlag, New York.

Cao, Z., Lee, D.-H., 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.

Codato, G., Fischetti, M., 2006. Combinatorial Benders' cuts for mixed-integer linear programming. Operations Research 54, 756–766.

Cordeau, J.-F., Soumis, F., Desrosiers, J., 2001. Simultaneous assignment of locomotives and cars to passenger trains. Operations Research 49, 531–548.

Geoffrion, A.M., Graves, G.W., 1974. Multicommodity distribution system design by Benders decomposition. Management Science 20, 822–844.

Hoogeveen, J.A., Lenstra, J.K., Veltman, B., 1996. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. European Journal of Operational Research 89, 172–175.

Hooker, J.N., 2000. Logic-Based Methods for Optimization: Combining Optimization and Constraints Satisfaction. John Wiley and Sons, Singapore.

Kim, K.H., Bae, J.W., 2004. A look-ahead dispatching method for automated guided vehicles in automated port container terminals. Transportation Science 38, 224–234.

Kim, K.H., Kim, K.Y., 1999a. An optimal routing algorithm for a transfer crane in port container terminals. Transportation Science 33, 17–33.

Kim, K.Y., Kim, K.H., 1999b. A routing algorithm for a straddle carrier to load export containers onto a containership. International Journal of Production Economics 59, 425–433.

Lee, D.-H., Cao, J.X., Shi, Q., 2008. Integrated model for truck scheduling and storage allocation problem at container terminal. In: Proceeding of TRB 87th Annual Meeting.

Lee, D.-H., Cao, Z., Meng, Q., 2007. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. International Journal of Production Economics 107, 115–124.

Ng, W.C., 2005. Crane scheduling in container yards with inter-crane interference. European Journal of Operational Research 164, 64–78.

Ng, W.C., Mak, K.L., 2005. Yard crane scheduling in port container terminals. Applied Mathematical Modeling, 263–276.

Ng, W.C., Mak, K.L., Zhang, Y.X., 2007. Scheduling trucks in container terminals using a genetic algorithm. Engineering Optimization 39, 33–47.

Stahlbock, R., Voß, S., 2008. Operations research at container terminals: a literature update. OR Spectrum 30, 1–52.

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