

THE INTEGRATED SCHEDULING PROBLEM IN CONTAINER TERMINAL WITH DUAL-CYCLE OPERATION

Zhang, Y.*; Rong, Z.*** & Liu, Z.-X.**

* Wuhan University of Technology, 1040 Heping Avenue, Wuhan, Hubei 430063, China

** Wuhan University of Science and Technology, 947 Heping Avenue, Wuhan, Hubei 430081, China

E-Mail: rongzhijun@263.net (° Corresponding author)

Abstract

The paper proposes an integrated scheduling problem for dual-cycle operation in container terminal, which can be described as a 3-stage hybrid flow shop problem with multi-job families and no-buffer. The integrated scheduling problem is formulated as a mixed-integer programming model. Due to the computational intractability, a simulation-based heuristic algorithm is developed for problem solution. State transition of yard truck, inventory and quota of quay crane and yard crane are introduced into the heuristic algorithm. Computational experiments and simulation analysis are conducted to evaluate the effectiveness of the proposed heuristic algorithm. The results show that the algorithm presented is very effective for the equipment scheduling of meta-container terminal with dual-cycle operation. (Received, processed and accepted by the Chinese Representative Office.)

Key Words: Container Operation System, Dual-Cycle Operation, Scheduling, Heuristics

1. INTRODUCTION

Fig. 1 shows a typical layout of container terminal [1]. A port operates as follows: (1) when a vessel docks and berths, quay crane (QC) picks up an inbound container from the containership and loads it onto a yard truck (YT). The YT then transports the container to a container yard and a yard crane (YC) in the yard picks off it from the YT and stacks it. There only exists inbound container flow; (2) after finishing all loading tasks, YC picks up an outbound container from stack and loads it onto a YT. The YT then transports the container to a QC, which will finally load the container onto the containership. There only exists outbound container flow. The above process is the first-unloading last-loading operation mode. Since there is always one-directional container flow, managers can arrange equipment scheduling and yard storage plan easily. But YT performing transportation task must make an empty movement to the quay area or to storage location in the yard. In order to improve efficiency of container terminal, more harbour bureaux start to use new handling technology which can reduce the empty movement of YT.

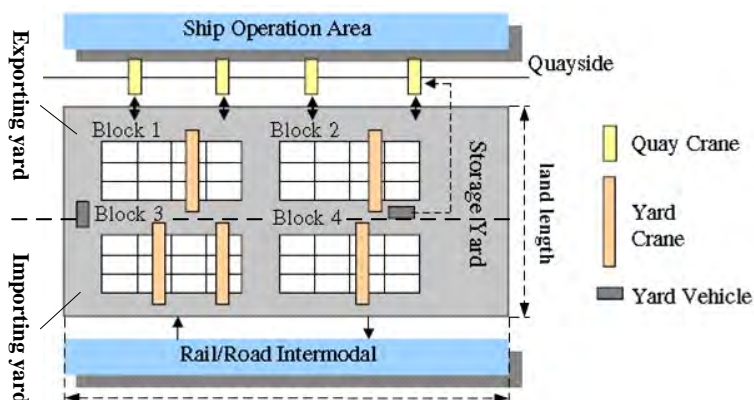


Figure 1: A typical layout of container terminal in China.

In recent years, the dual-cycle operation has been extended. Fig. 2 shows the process of dual-cycle operation [2]. When an YT transporting the outbound container reaches the quayside, QC picks off the container from the YT and loads it onto the planned slot on the containership; and then the QC discharges an inbound container from vessel's current bay and loads it onto the YT. The YT's efficiency can be doubled in the dual-cycle operation, because of the reduction of empty movement of YTs. The dual-cycle operation brings higher requirements for the scheduling of YTs, stowage of vessel and cooperation among equipment due to inbound and outbound container flow existing simultaneously. In this paper, we only pay attention to the integrated yard truck, quay crane and yard crane scheduling problem in container terminal with dual-cycle operation.

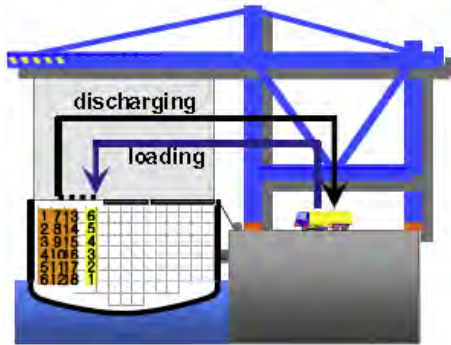


Figure 2: Dual-cycle operation.

Studies associated with production organization in a container terminal are concerned on berth allocation, container stowage, container deployment, slot allocating in yard, equipment deploying and scheduling, traffic flow controlling, description and evaluation of logistics system in the terminal. Bierwirth [3], Stahlbock [4], Steenken [5] and Vis [6] gave comprehensive reviews of studies above. Here, we provide only a brief review of previous studies related to the integrated scheduling problem of container terminal operation system and equipment scheduling of container operation system using dual cycle operation.

There are few literatures focused on the integrated scheduling problem of various types of handling equipment used at container terminals. Chen [1, 7] described container operation system as 3-stage hybrid flow shop, and she solved the integrated scheduling problem of multi-type equipment with tabu search (TS). Zeng [8] also used a 3-stage hybrid flow shop to simulate the operating system of container terminals, and adopted simulation optimization to solving the integrated scheduling problem. The optimization method was a mixed algorithm of neural network (NN) and GA. NN was mainly used for predicting individual adaptive function. Although literatures above focused on the integrated scheduling problem of various types of equipment in container terminal, they only concentrated on it in the operation mode of first-unloading last-loading or loading outbound containers, without taking dual-cycle operation mode into account.

As to the dual-cycle operation, Goodchild [9] described the double-cycling problem, and gave a solution in order to reduce the number of operations and operating time. Zhang [2] proposed a mixed integer programming mode for discharging and loading containers in a ship-bay, which was equivalent to maximizing the number of dual cycle operations. Meisel [10] adopted dual-cycle mode to minimize the inbound and outbound containers operating time in the bay by optimizing the operating sequence of QCs for any bay. These studies mainly focused on the containers operating sequence problem, and the objective was to improve the efficiency of QCs and to shorten the turnaround time of ships. However, the integrated scheduling problem of various types of equipment in container terminal using dual-cycle mode still was not considered.

In container terminal, the dual-cycle operation can decrease the time of empty movements of YT by making the YT perform the transporting tasks of inbound and outbound containers alternately. There is no buffer between QCs and YTs or YTs and YCs, thus the equipment scheduling is very complicated in container terminal using dual-cycle mode. Once one certain equipment is not available, other types of equipment are forced to wait. For example, if YT is not available, QC or YC need to wait; QC or YC are not available, YT needs to stand in a queue. That means QCs and YCs must cooperate with YTs to perform the loading and discharge task alternately.

This paper is organized as follows. In Section 2, the integrated scheduling problem of the container operation system using dual-cycle operation is described as three-stage hybrid flow shop problem with no-buffer and multi-job families. Section 3 constructs the mathematical model of the problem. Section 4 develops a heuristic algorithm to solve the problem. Section 5 compares the heuristic algorithm to Chen’s [1] and Zeng’s [8], and gives computational experiments of the heuristic algorithm based on cases of real world from mega container terminal. Finally, Section 6 presents the conclusions.

2. DESCRIPTION OF INTEGRATED SCHEDULING PROBLEM

Hybrid flow shop (HFS) is an extension of flow shop which exists extensively in the real world such as electrical manufacturing (Choi [11]), iron industry (Atighehchian [12]), textile industry (Montoya-Torres [13]) and ports operation (Chen [1, 7]; Zeng [8]). For the standard HFS, Ruiz [14] defined it as follows:

The number of processing stages m is at least 2. Each stage k has $m(k) \geq 1$ machines in parallel and in at least one of the stages $m(k) \geq 1$. All jobs are processed following the same production flow: stage 1, stage 2... stage m . A job might skip any number of stages provided it is processed in at least one of them. Each job j requires a processing time p_{jk} in stage k . We shall refer to the processing of job j in stage k as operation O_{jk} .

A standard HFS problem assumes that the storage capacity of buffers between stages is infinite. However, under the real production environment, such as iron industry, petrochemical industry and port operation, equipment scheduling problem is usually HFS scheduling problem with limited or no buffer. The HFS scheduling problem with limited or no buffer is more difficult to deal with than the standard HFS scheduling problem. This HFS problem with finite or no-buffer characteristics are often called HFS problem with blocking (HFS-B).

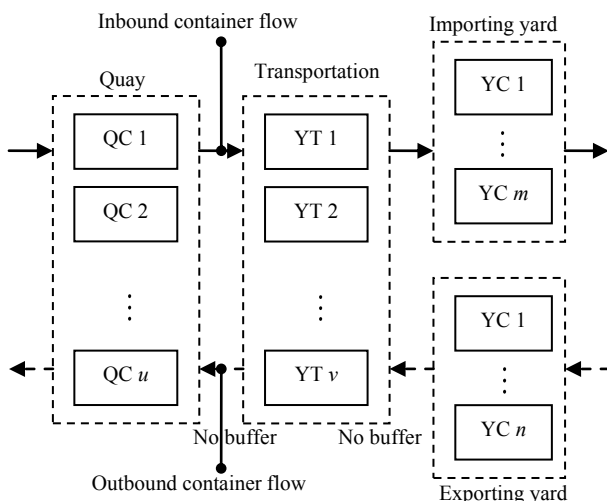


Figure 3: Work flow of container operation system using dual cycle operation.

The integrated scheduling problem of various types of equipment in container terminal using dual-cycle mode is a typical HFS problem with multi-job families and no-buffer. Fig. 3 shows the process with dual cycle. In Fig. 3, there are two flows: inbound container flow and outbound container flow. Every container in same flow follows same sequence during the 3 stages. All containers in one flow are processed following the same production flow. Each stage has several unrelated parallel machines and there are no buffers between stages. During the second stage (horizontal transportation), setup time and operating time related to containers sequence and to upstream and downstream machines should be considered. Considering what mentioned above and combining with the definition and characteristics of HFS problem provided by Ruiz [14] and Ribas [15], we treat the containers in two different flows as two-job families and present a 3-stage HFS problem with no-buffer and multi-job families whose process is showed in Fig. 4.

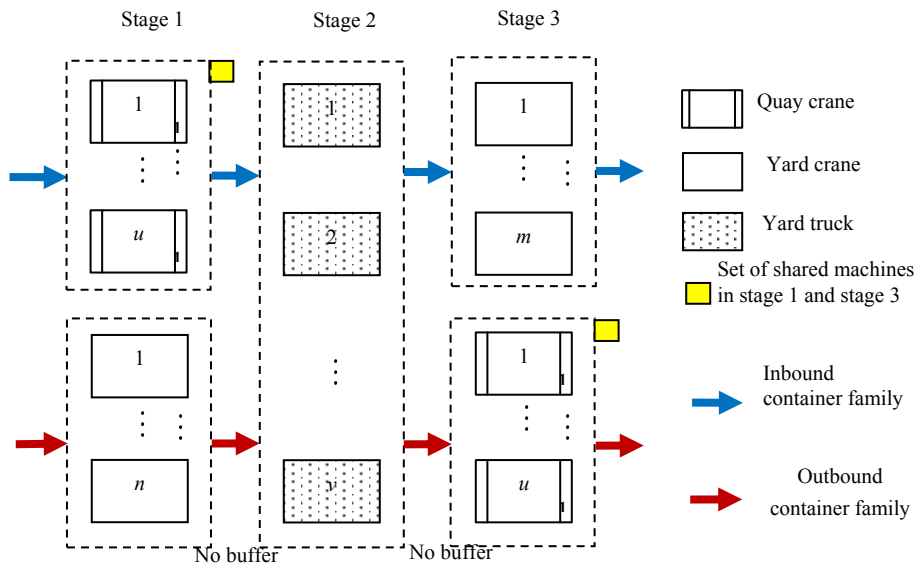


Figure 4: Three-stage HFS problem with no-buffer and multi-job families.

In Fig. 4, the characteristics of HFS problem with no-buffer and multi-job families are as follows: (1) In view of different job families, first stage and third stage have different dedicated machines respectively serving for jobs from two different job families; (2) For one job in a certain job families (inbound job family or outbound job family), it goes through three stages and each stage has some unrelated parallel machines for the job to choose; (3) There is no buffer between two successive stages; (4) Part of machine sets have intersection at the first stage and the third stage, which will increase the competition for resources; (5) As the processing time is related with machines and sequence, the processing time p_{i2} of job i at the second stage is not known in advance, which is proportional to the distance between upstream machine $m(O_{i1})$ and downstream machine $m(O_{i3})$. O_{i1} and O_{i3} represent the operation of the job i at the first stage and the third stage. (6) The setup time S_{i2} of job i at the second stage is not known in advance. If operation O_{k2} immediately precedes O_{i2} , S_{i2} can also represent the setup time S_{k2i2} between O_{k2} and O_{i2} . S_{k2i2} is proportional to the distance between $m(O_{k3})$ and $m(O_{i1})$. Setup time is related with machines and sequence.

For the HFS problem with multi-job families and no buffer, it becomes much complicated to construct a model to solve it compared with typical HFS problem. Chen [1, 7] and Zeng [8] had already validated the complication of typical HFS problem under containers operation circumstance. However, HFS problem with multi-job families and no-buffer is different from their researches because they only considered one flow or two flows in first-unloading last-loading mode without considering these characteristics: (1), (4), (5) and (6). While HFS

problem with multi-job families and no buffer has some new characteristics and is more complication, it can be extended on the foundation of the model developed by Chen [1].

3. THE MATHEMATICAL MODEL OF PROBLEM

This section presents a mixed-integer programming model for the HFS problem with multi-job families and no buffer. The following notation is used to formulate the problem.

Notation: i, k – job index. j, l – stage index, $j, l = 1, 2, 3$. m – machine index. N^- – the set of jobs of inbound job family. N^+ – the set of jobs of outbound job family. N – the set of all the jobs, $N = N^- \cup N^+$. P – the set of ordered pairs of jobs between which there is a precedence relationship, when job i must precede job k , $\forall (i, k) \in P$. O_{ij} – operation of job i at stage j , each job has three operations. M_j^-, M_j^+ – the set of machines which process inbound job family and outbound job family respectively in stage j . M_j – the set of machines at stage j , $M_j = M_j^- \cup M_j^+$, $j \in \{1, 2, 3\}$. QC, YT, YC^-, YC^+ – the set of QCs, YTs, YCs of export storage yard, and YCs of import storage yard respectively. $\exists M_2^- = M_2^+ = YT, M_1^- = M_3^+ = QC, M_1^+ = YC^+, M_3^- = YC^-$. p_{ij}^-, p_{ij}^+ – processing time of operation O_{ij} when $i \in N^-$ and $i \in N^+$ respectively. τ_{sd} – traveling time of yard truck from machine s to machine d , which is known in advance. H, H_0 – sufficiently large constant.

Decision variables: $x_{ijm} = 1$, if O_{ij} is assigned to machine m ; 0, otherwise. $y_{ijklm} = 1$, if O_{ij} and O_{kl} are assigned to the same machine m ; 0, otherwise. $u_{ijklm} = 1$, if O_{ij} precedes O_{kl} (not necessarily immediately) on machine m ; 0, otherwise. $z_{ijklm} = 1$, if O_{ij} immediately precedes O_{kl} on machine m ; 0, otherwise. $w_{im_1m_3} = 1$, if O_{i1} and O_{i3} operate respectively on machine $m_1 \in M_1$ and $m_3 \in M_3$; 0, otherwise. $v_{m_2im_3km_1} = 1$, if O_{i2} immediately precedes O_{k2} on machine $m_2 \in M_2$ and O_{i3}, O_{k1} operate respectively on machine $m_3 \in M_3$ and $m_1 \in M_1$; 0, otherwise. p_{ij} – processing time of operation O_{ij} , $j = 2$. s_{ijkl} – setup time between O_{ij} and O_{kl} , $j = l = 2$. t_{ij} – the starting time of O_{ij} .

The integrated scheduling problem may be formulated as follows:

$$\text{Minimize } C_{\max} = \min \{ \max_{ij} (t_{ij} + p_{ij}) \} \quad (1)$$

Subject to

$$t_{ij} \geq 0, \forall i \in N, \forall j \in \{1, 2, 3\} \quad (2)$$

$$t_{ij} + p_{ij} \leq t_{i(j+1)}, \forall i \in N, \forall j \in \{1, 2\} \quad (3)$$

$$\sum_{m \in M_j^-} x_{ijm} = 1, \forall i \in N^-, \forall j \in \{1, 2, 3\} \quad (4)$$

$$\sum_{m \in M_j^+} x_{ijm} = 1, \forall i \in N^+, \forall j \in \{1, 2, 3\} \quad (5)$$

$$y_{ijklm} \leq 0.5(|1 - |j - l||)(x_{ijm} + x_{klm}) \leq (y_{ijklm} + 0.5), \forall i, k \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (6)$$

$$y_{ijklm} = y_{klijm}, \forall i, k \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (7)$$

$$u_{ijklm} + u_{klijm} = y_{ijklm}, \forall i, k \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (8)$$

$$u_{ijklm} - z_{ijklm} \geq 0, \forall i, k \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (9)$$

$$\sum_{k \in N} z_{ijklm} \leq 1, \forall i \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (10)$$

$$\sum_{k \in N} z_{klijm} \leq 1, \forall i \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (11)$$

$$x_{ijm} \geq 0.5(\sum_{k \in N} z_{ijklm} + \sum_{k \in N} z_{klijm}) \geq x_{ijm} - 0.5, \forall i \in N, \forall j, l \in \{1, 2, 3\}, \forall m \in M_j \quad (12)$$

$$t_{i1} \leq t_{k1}, \forall (i, k) \in P \quad (13)$$

$$t_{i(j+1)} + s_{ijkl} \leq t_{kl} + H(1 - z_{ijklm}), \forall i, k \in N, \forall j \in \{1, 2\}, \forall l \in \{1, 2, 3\}, \forall m \in M_j \quad (14)$$

$$(t_{ij} + p_{ij}) + s_{ijkl} \leq t_{kl} + H(1 - z_{ijklm}), \forall i, k \in N, j = 3, \forall l \in \{1, 2, 3\}, \forall m \in M_j \quad (15)$$

$$w_{im_1m_3} \leq 0.5(x_{i1m_1} + x_{i3m_3}) \leq (w_{im_1m_3} + 0.5), \forall i \in N, \forall m_1 \in M_1, \forall m_3 \in M_3 \quad (16)$$

$$p_{ij} = \begin{cases} p_{i1}^- & i \in N^-, j = 1 \\ p_{i1}^+ & i \in N^+, j = 1 \\ \sum_{m_1 \in M_1} \sum_{m_3 \in M_3} \tau_{m_1m_3} w_{im_1m_3} & \forall i \in N, j = 2 \\ p_{i3}^- & i \in N^-, j = 3 \\ p_{i3}^+ & i \in N^+, j = 3 \end{cases} \quad (17)$$

$$3v_{m_2im_3km_1} \leq (x_{i3m_3} + x_{k1m_1} + z_{i2k2m_2}) \leq (3v_{m_2im_3km_1} + 2), \forall i, k \in N, \forall m_j \in M_j, j = 1, 2, 3 \quad (18)$$

$$s_{ijkl} = \begin{cases} \sum_{m_1 \in M_1} \sum_{m_2 \in M_2} \sum_{m_3 \in M_3} \tau_{m_1m_3} v_{m_2im_3km_1} & j = l = 2 \\ \sum_{m \in M_1 \cup M_2 \cup M_3} (1 - z_{ijklm}) H_0 & else \end{cases}, \forall i, k \in N, \forall j, l \in \{1, 2, 3\} \quad (19)$$

In the objective function (1), the makespan is minimized. Constraint (2) guarantees that the starting time of every operation is not smaller than 0. Constraint (3) ensures the order of operations for each container. Constraints (4) and (5) ensure that every operation must be completed by exactly one machine, and they define the type of machine processing jobs of inbound job family or outbound job family. Constraints (6) and (7) define $y_{ijklm} = y_{kljim} = 1$ when $|j - l| \neq 1$ and $x_{ijm} = x_{klm} = 1$. When O_{ij} and O_{kl} are assigned to the machine m , constraint (8) defines u_{ijklm} such that either u_{ijklm} or u_{kljim} is equal to 1. Constraint (9) defines $z_{ijklm} = 1$ when O_{ij} immediately precedes O_{kl} . Constraints (10) or (11) respectively ensure that each operation O_{ij} has, at most, one predecessor or one successor on machine m . Constraint (12) is a flow balance constraint, guaranteeing that jobs are handled in well-defined sequences. Constraint (13) denotes that job i precedes job k at stage 1. Constraints (14) and (15) determine the starting time of O_{kl} and blocking requirement. When $z_{ijklm} = 1$, machine m is available for O_{kl} only after job i leaves machine m and machine m is ready. Constraint (16) defines $w_{im_1m_3} = 1$, when $x_{i1m_1} = x_{i3m_3} = 1$, otherwise, $w_{im_1m_3} = 0$. Constraint (17) defines processing time related to machines and sequence. Constraint (18) defines $v_{m_2im_3km_1} = 1$ when $x_{i3m_3} = x_{k1m_1} = z_{i2k2m_2} = 1$, otherwise, $v_{m_2im_3km_1} = 0$. Constraint (19) defines setup time related to machines and sequence.

The small scale mixed integer programming model can be solved with CPLEX and AMPL. But in real port operation, the number of jobs is more than 1000 and the amount of machines in three stages is more than 100. It is doomed unable to obtain optimal solution for large-scale problem in an acceptable computational time. Thus, we need develop algorithms with high quality and efficiency to solve the HFS problem with multi-job families and no buffer.

4. THE SOLUTION APPROACH

Before designing the heuristic algorithm solving the HFS problem with multi-job families and no buffer, we introduce some concepts as follows:

(1) Machines (crane) quota: the remained capacity of a machine to process jobs. This concept only suits these machines in the first and the third stage in Fig. 4. Every machine has two types of quota respectively corresponding to inbound job family (A) and outbound job family (B). When the remained capacity (quota) of machine falls to zero, it means jobs cannot be allocated to the machine. For example, if the initial quotas of a machine for A and B both are 100 and after a period, the machine processes 70 A family jobs and 30 B family jobs, then the machine can still process 30 A family jobs but B family jobs cannot be allocated to it. Once the machine finishes processing 30 A family jobs, it turns into inactive state.

(2) Machines (crane) inventory: the inventory of a machine represents how busy it is (Briskorn [16]). This concept only suits these machines in the first and the third stage in Fig.

4. The inventory of a machine is equal to the number of YTs queuing at the machine (crane) plus the number of en route YTs dispatched to the machine (crane). The smaller the inventory of a machine is, the greater the risk is for machine starvation. That is, the machine is more likely to become idle. As there is no buffer, block will exist between stage 1 and stage 2, stage 2 and stage 3. This means these machines (cranes) in stage 1 and stage 3 always need these machines (YTs) in stage 2 to cooperate with them.

(3) State transition: describe the state transition of machines (YTs) in stage 2. For any machine $m \in M_2$ at stage 2 in HFS-B, once available, it will receive containers transportation task. After being allocated with tasks, the yard truck has to make empty movement from current position to the quay or yard. This trip time is setup time of yard truck. When the yard truck arrives at the crane, it will queue up and be loaded by the crane, and then make a trip to its destination. After arriving at its destination, it will queue up and be unloaded by a crane. As yard truck servicing both quay cranes and yard cranes, it always circularly goes through these states: idle (receiving task) \rightarrow preparing (empty movement) \rightarrow waiting (queuing up) and loading \rightarrow transporting (loaded movement) \rightarrow waiting (queuing up) and discharging \rightarrow idle (sending demand).

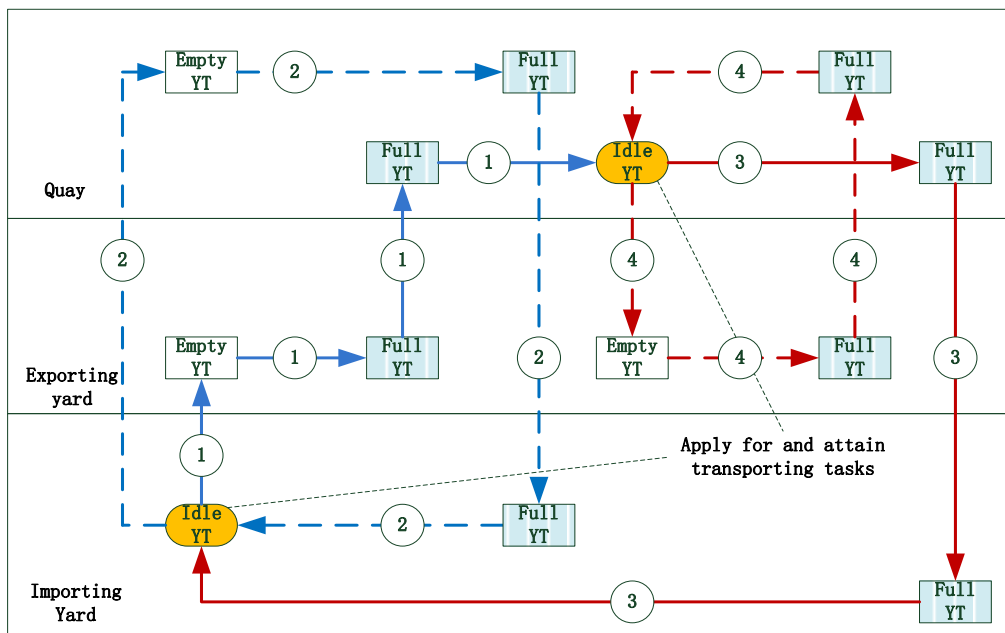


Figure 5: State transition of YT.

Fig. 5 shows state transition of machine (YT) in HFS-B. When an YT is available, the YT's state is idle state denoted by "oval-shape". It will apply for and obtain container transporting task. In Fig. 5, when the idle YT is at quayside denoted by "Y", it will give priority to an inbound container transporting task denoted by "circle-shape 3" or secondly accept an outbound container transporting task denoted by "circle-shape 4". When the idle YT is at importing yard denoted by "Z", it will give priority to an outbound container transporting task denoted by "circle-shape 1" or secondly accept an inbound container transporting task denoted by "circle-shape 2". Every task has a source and a sink, respectively corresponding to the machine (crane) loading container into the YT executed the task and the machine (crane) unloading container from the YT executed the task. At source or sink of the task, the state of the YT will be changed from "no-load" to "full-load" or from "full-load" to "no-load". Since combination tasks "1" \rightarrow "3" or "3" \rightarrow "1" decrease empty movement of YT and the waiting time of crane, we can construct heuristic algorithm by judging the position of idle YTs and assigning combination tasks ("1" \rightarrow "3" or "3" \rightarrow "1") to idle YTs as possible.

According to what mentioned above, we can give the description of procedure of heuristic algorithm: firstly, obtain the set of idle YTs in simulation procedure constructed with Plant Simulation Software; assign combination tasks to idle YTs according to task assignment sub-procedure; after every idle YT has obtained task, simulation procedure drives YT moving from the source to sink of the task assigned to the YT; when a YT is loaded or unloaded, simulation procedure immediately updates information of quotas and inventories of machines (cranes) and state of machines (YTs).

Let us introduce the used notation: $C = C^- \cup C^+ = N$, union set of inbound and outbound containers. S – the set of idle YTs. $\psi(m)$ – the inventory of the machine $m \in M_1 \cup M_3$, $L(m \in M_2) = (m_s, m_d)$, the route determined by source m_s and sink m_d of the container transporting task allocated to the machine (YT) $m \in M_2$. Here, m_s or m_d represent the crane loading or unloading YT m . $\omega(m)$ – the quota of machine m , $m \in M_1 \cup M_3$. $Q(m \in M_1) \subset C$ – the set of task sequence of machine m . According to the requirement of sequence of loading or unloading a ship, $Q(m)$ is known in advance. Y and Z^- represent location set of quayside and importing yard respectively.

The procedure of heuristic algorithm is as follows:

-
1. Let $S = \phi$, and determine the states of all machines at stage 2. If $m \in M_2$ is idle, then $S = S \cup \{m\}$.
 2. If $S \neq \phi$, pick up the first element m^* from S and get its current location $p(m^*)$; otherwise, go to step 3.
 - 2.1 Call task assignment sub-procedure ($p(m^*)$) which can assign container transporting task k and route $L(m^*)$ to m^* .

Let $C = C \setminus \{k\}$, $S = S \setminus \{m^*\}$.
 - 2.2 If $S \neq \phi$, go to step 2; otherwise, go to step 3.
 3. Run simulation procedure constructed with Plant Simulation Software.
 - 3.1. $\forall m \in M_2$, update the state and location of m ; if m is idle, let $S = S \cup \{m\}$.
 - 3.2. $\forall m \in M_1 \cup M_3$, when machine m loads or unloads one YT, $\psi(m) --$.
 - 3.3. If $S \neq \phi$, go to step 2.
 - 3.4. If $C \neq \phi$, go to step 3; otherwise, go to step 4.
 4. Output data and stop the simulation.
-

In the procedure of heuristic algorithm, the task assignment sub-procedure is used to construct combination task “1” \rightarrow “3” or “3” \rightarrow “1” for every idle machine (YT) $m \in S \subset M_2$, so as to decrease the empty move of YT. For every sub-task (“1”, “2”, “3” and “4”) of combination task allocated to idle YT, we need to determine the source and the sink of the sub-task. The source m_s and the sink m_d are respectively corresponding to the machine (crane) $m \in M_1 \cup M_3$ loading the YT and the machine (crane) $m \in M_1 \cup M_3$ unloading the YT. We select a machine $m \in M_1 \cup M_3$ as the source or the sink of sub-task of one combination task by sequencing these machines ($\forall m \in M_1 \cup M_3$) in the non-decreasing order of inventories or by ordering them in non-increasing order of quotas when inventories of machines have same value. The detail of task assignment sub-procedure is as follows:

-
1. According to $p(m^*)$, determine the task type $\tau(m^*)$ of the machine m^* .

If $p(m^*) \in Z^-$ & $C^+ \neq \phi$, let $\tau(m^*) = “1”$;

Else If $p(m^*) \in Z^-$ & $C^+ = \phi$ & $C \neq \phi$, let $\tau(m^*) = “2”$;

Else If $p(m^*) \in Y$ & $C^- \neq \phi$, let $\tau(m^*) = “3”$;

Else If $p(m^*) \in Y$ & $C^- = \phi$ & $C \neq \phi$, let $\tau(m^*) = “4”$;

Else let machine (YT) m^* to yard truck pool;

End.
-

2. According to $\tau(m^*)$, determine the machine type of m_s and m_d of $L(m^*) = (m_s, m_d)$.
 If $\tau(m^*) = "1"$ or $"4"$, let $m_s \in M_1^+$ and $m_d \in M_3^+$;
 Else If $\tau(m^*) = "2"$ or $"3"$, let $m_s \in M_1^-$ and $m_d \in M_3^-$.
3. Repeat 3.1 ~ 3.3 until the two machines corresponding to m_s and m_d are determined.
 - 3.1 Let l denote source m_s or sink m_d .
 If $l \in M_1^+$, let $M_l = M_1^+$;
 Else If $l \in M_3^+$, let $M_l = M_3^+$;
 Else If $l \in M_1^-$, let $M_l = M_1^-$;
 Else If $l \in M_3^-$, let $M_l = M_3^-$;
 End.
 - 3.2 Sequence machines ($\exists m \in M_l$) in non-decreasing order of inventory $\psi(m \in M_l)$ or in non-increasing order of quota $\omega(m \in M_l)$ when machines ($\exists m \in M_l$) have same inventory.
 - 3.3 Select the first element j from M_l , and let $l = j$, $\psi(j)++$ and $\omega(j)--$.
4. Determine task k according to machine m_s .
 Select the first element r of $Q(m_s)$, and let $k = r$.
5. Output container transporting task k and route $L(m^*)$ assigned to the machine m^* .

5. PERFORMANCE EVALUATION OF ALGORITHM

In this section, we evaluate the performance of our algorithms by comparing them with other algorithms. We use 20 scenarios from Chen [1] shown in Table I. The operation times of machines are generated from uniform distribution of U(105, 161) for the quay crane operation, U(9, 41) for the yard crane picking operation, U(38, 70) for the yard crane dropping operation and U(60, 130) for the yard truck transferring operation.

Table I: Performance comparison of HA-S with TS1 for small-size scenarios.

Scenario	Problem size					HA-S				TA1	
	QC	YC	YT	N ⁻	N ⁺	Makespan	η_{qc}	η_{yc}	Gap	Gap	CPU time (s)
1	2	4	4	10	10	1735.65	76.48	22.77	16.40	0.71	26.0
2	2	4	6	10	10	1464.60	90.83	27.23	3.27	1.3	32.7
3	2	4	6	20	20	2792.27	95.04	28.39	1.65	0.15	34.0
4	2	4	8	20	20	2716.55	97.92	29.18	0.25	1.1	39.5
5	2	6	8	20	20	2724.76	97.68	19.42	0.41	0.45	34.0
6	2	4	8	40	40	5379.47	98.78	29.41	0.24	0.52	43.0
7	2	6	8	40	40	5377.80	98.80	19.58	0.27	0.38	46.7
8	2	6	10	40	40	5378.19	98.76	19.62	0.10	0.33	48.0
9	2	6	8	50	50	6713.79	98.81	19.64	0.41	0.48	95.2
10	2	6	10	50	50	6716.08	98.82	19.63	0.23	0.44	94.0
11	4	6	10	50	50	3693.91	89.79	35.63	6.06	9.39	131.7
12	4	8	10	50	50	3709.80	89.52	26.65	6.39	8.09	113.3
13	4	8	12	60	60	4155.58	95.83	28.55	1.77	8.81	197.7
14	4	8	15	60	60	4104.58	97.01	28.93	1.66	8.36	179.0
15	4	10	12	60	60	4157.22	95.75	22.85	1.64	7.58	216.5
16	4	10	15	60	60	4104.05	97.06	23.12	1.74	8.63	112.5
17	4	10	15	80	80	5443.07	97.66	23.26	1.38	8.43	272.3
18	4	10	20	80	80	5425.84	97.97	23.37	1.11	8.26	269.0
19	4	10	20	100	100	6748.96	98.44	23.48	0.82	8.72	496.2
20	4	12	24	100	100	6751.30	98.47	19.54	1.00	8.58	563.0

η_i denotes equipment utilization ratio. N⁻ (N⁺) denotes number of import (export) containers.

For every scenario, we randomly generate 100 instances, run them in dual-cycle operation circumstance, and record the gap (%) of the heuristic makespan (second) from lower bound over these 100 instances. Here, the lower bound is developed from [17]. The CPU time of our heuristic algorithm in all scenarios is not greater than 1 second. Table I gives performance comparison of our heuristic algorithm (denoted HA-S) with TA1 (Chen [1]), which is a tabu search algorithm with MIH_ND (multiple insertion heuristics with non-delay).

In Table I, the gap performance of HA-S is weaker than TA1 at scenarios 1~3, while the gap performance of HA-S is better than TA1 at scenarios 4~20. This is because: (1) at scenarios 1~3, loading and unloading tasks are less, and simulation is still at warm-up stage; (2) quay crane and yard crane both are bottleneck machines, and the core of equipment scheduling in container terminal is to balance workload of these cranes. Balancing workload needs adequate loading and unloading tasks and certain amount of equipment (QC, YC and YT). In Table I, apparently, scenarios 1~3 are lack of equipment or tasks. With the problem size increasing, such as case 13 to case 20, the performance of HA-S tends to be steady and better than TA1.

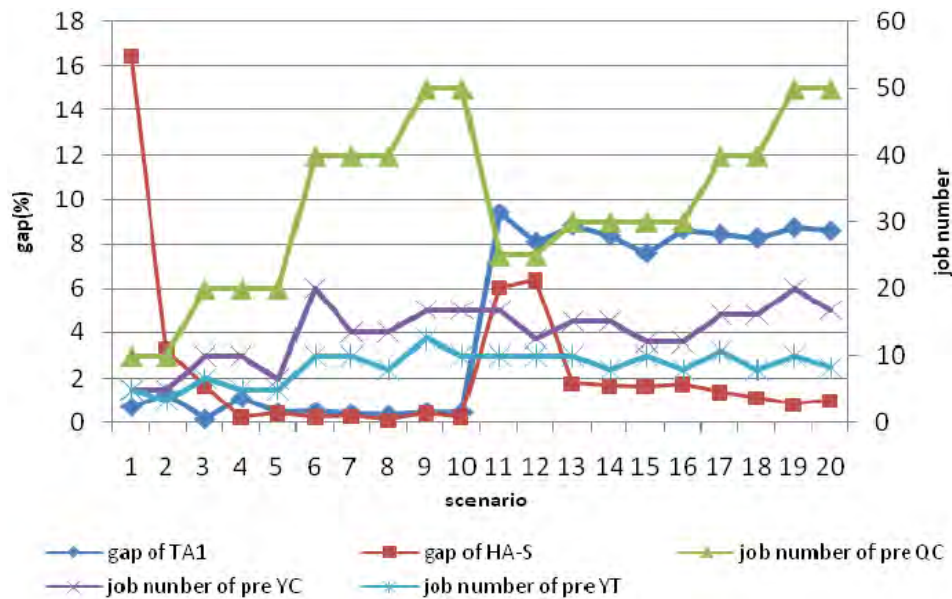


Figure 6: Gap changing trend of heuristic algorithm and HA-S with tasks changing.

Besides, in Fig. 6, it is obvious that the relationship between the job number of pre QC and the gap feature of HA-S is reciprocal. Therefore the problem size of scenario has great impact on the gap of algorithms.

Due to the feature of small problem size in Table I, we further evaluate HA-S compared with GN-LPT and TA1 in medium-size scenarios. In Table II, Zeng [8] gave the comparison between GN-LPT himself and TA1 for medium-size scenarios. GN-LPT represents: Initialize container sequence according to LPT (longest processing time first) rule, improve the sequence by genetic algorithm, and predict objective function and filter out bad solution by neural network.

Table II: Performance comparison of HA-S with GN-LPT and TA1 for medium-size scenarios.

Scenario	Problem size					Gap (%)			CPU time (s)		
	QC	YC	YT	N ⁻	N ⁺	HA-S	TA1	GN-LPT	HA-S	TA1	GN-LPT
21	4	10	16	200	200	6.04	4.01	2.63	<2	635.4	1053.6
22	4	10	16	400	400	3.39	6.17	3.35	<2	1764.2	2321.7
23	4	10	16	500	500	2.74	8.04	4.07	<2	2846.5	2930.4

In Table II, the gap performance of HA-S is weaker than TA1 and GN-LPT in scenario 21. But with the amount of jobs increasing, the gap of HA-S is gradually descending, and is better than TA1 and GN-LPT in scenario 23; yet gaps of TA1 and GN-LPT are gradually ascending with the amount of jobs increasing. In a mega maritime container terminal, the amount of the jobs/containers is usually greater than 1000, so HA-S can achieve good performance with the increasing of the amount of jobs in an actual maritime container terminal.

Therefore we further construct four big-size scenarios shown in table III, and the details of the computational experiments are shown in [18]. The operating times of quay crane, yard crane and the transferring time of yard truck are from the survey of actual terminal of Shanghai in China, and the details are shown in [18]. The quay length and land length are 1000 meter and 800 meter respectively.

Table III: Basic information of big-size scenarios.

Scenario	Problem size				
	QC	YC	YT	N ⁻	N ⁺
24	6	12	18~120	1200	1200
25	6	24	18~120	1200	1200
26	12	24	18~120	2400	2400
27	12	48	18~120	2400	2400

Figs. 7 and 8 give average utilization of QCs and gaps for four big-size scenarios. In Fig. 7, we can conclude that the sensible number of YTs for scenarios 24~27 are 30, 36, 60 and 72 respectively, because above numbers are corresponding to the x-coordinate values of knee points of curves in Fig. 7. Apparently, adding the number of YTs cannot obviously improve utilization of QCs when the numbers of YTs for scenario 24~27 are bigger than the x-coordinate values of knee points of curves in Fig. 7, but the cost of buying and maintaining extra YTs is very expensive. When we use sensible YTs fleet size for scenario 24~27 to view Fig. 8, the computational experiments tell us a truth that HA-S algorithm can keep away from the worst gap because of diminishing returns between the yard truck fleet size and gap. The worst gap occurs at the left of the knee point corresponding to a yard truck fleet size, and the sensible yard truck fleet size is always at the right of the knee point. So HA-S algorithm presented by us can show good robustness in actual port operation.

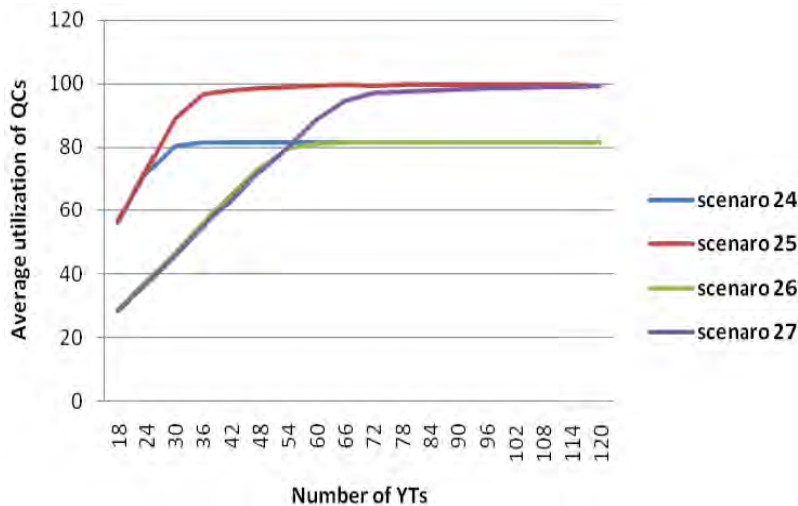


Figure 7: Utilization of QCs for four big-size scenarios.

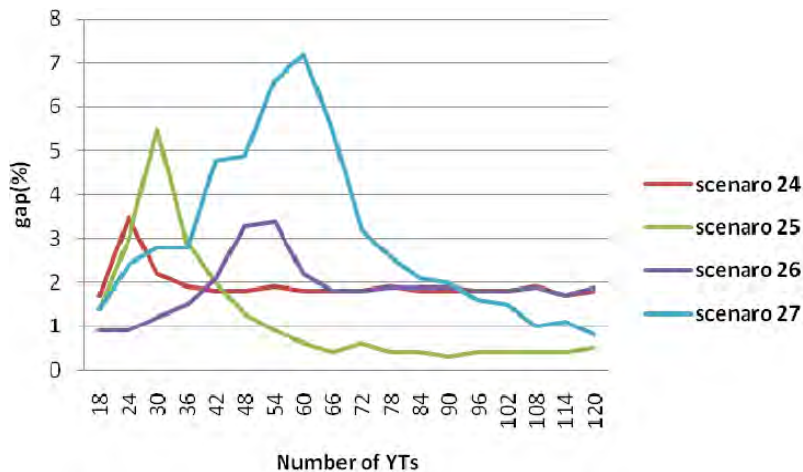


Figure 8: Gaps of four big-size scenarios.

6. CONCLUSION

In this paper, the integrated scheduling problem of container operation system in a maritime terminal using dual cycle operation is described as 3-stage hybrid flow shop problem with multi-job families, blocking, sequence-dependent setup time and operating time. Here, the setup time and the operation time of each machine at stage 2 depend on sequence and location of machine in upstream and downstream, and multi-job families increase competition for resources (i.e., quay crane). Combined with these characteristics, we construct a mixed integer programming model. Since HFS problem is NP-hard, we develop a simulation-based heuristic algorithm (HA-S) based on state transitions of YTs, crane's inventory and quota. The HA-S consists of heuristic algorithm and task assignment sub-procedure: heuristic algorithm drives simulation model based on state transitions of machines in stage 2, and calls task assignment sub-procedure to allocate machines in stage 1 and 3 to idle machines in stage 2 based on quota and inventory of machines in stage 1 and 2. The HA-S is compared with TA1 and GN-LPT for small-size and medium-size scenarios, and shows good performance that the gap of HA-S's makespan from lower bound decreases when number of tasks allocated to quay crane increases. We also introduce big-size scenarios, and HA-S can keep away from the worst performance of gap because of the truth of diminishing returns between the yard truck fleet size and the gap. Hence, the algorithm is very effective for the integrated scheduling problem of container operation system using dual-cycle operation in mega container terminal.

7. ACKNOWLEDGEMENT

The authors acknowledge that the research is supported by National Natural Science Fund of China (71372202, 51175388) and Fundamental Research Funds for the Central Universities (WUT: 2013-IV-057).

REFERENCES

- [1] Chen, L.; Bostel, N.; Dejax, P.; Cai, J.; Xi, L. F. (2007). A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal, *European Journal of Operational Research*, Vol. 181, No. 1, 40-58, doi:10.1016/j.ejor.2006.06.033
- [2] Zhang, H.; Kim, K. H. (2009). Maximizing the number of dual-cycle operations of quay cranes in container terminals, *Computers & Industrial Engineering*, Vol. 56, No. 3, 979-992, doi:10.1016/j.cie.2008.09.008

- [3] Bierwirth, C.; Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals, *European Journal of Operational Research*, Vol. 202, No. 3, 615-627, [doi:10.1016/j.ejor.2009.05.031](https://doi.org/10.1016/j.ejor.2009.05.031)
- [4] Stahlbock, R.; Voß, S. (2008). Operations research at container terminals: a literature update, *OR Spectrum*, Vol. 30, No. 1, 1-52, [doi:10.1007/s00291-007-0100-9](https://doi.org/10.1007/s00291-007-0100-9)
- [5] Steenken, D.; Voß, S.; Stahlbock, R. (2004). Container terminal operation and operations research – a classification and literature review, *OR Spectrum*, Vol. 26, No. 1, 3-49, [doi:10.1007/s00291-003-0157-z](https://doi.org/10.1007/s00291-003-0157-z)
- [6] Vis, I. F. A.; de Koster, R. (2003). Transshipment of containers at a container terminal: An overview, *European Journal of Operational Research*, Vol. 147, No. 1, 1-16, [doi:10.1016/S0377-2217\(02\)00293-X](https://doi.org/10.1016/S0377-2217(02)00293-X)
- [7] Chen, L.; Langevin, A.; Lu, Z. Q. (2013). Integrated scheduling of crane handling and truck transportation in a maritime container terminal, *European Journal of Operational Research*, Vol. 225, No. 1, 142-152, [doi:10.1016/j.ejor.2012.09.019](https://doi.org/10.1016/j.ejor.2012.09.019)
- [8] Zeng, Q. C.; Yang, Z. Z. (2009). Integrating simulation and optimization to schedule loading operations in container terminals, *Computers & Operations Research*, Vol. 36, No. 6, 1935-1944, [doi:10.1016/j.cor.2008.06.010](https://doi.org/10.1016/j.cor.2008.06.010)
- [9] Goodchild, A. V.; Daganzo, C. F. (2006). Double-cycling strategies for container ships and their effect on ship loading and unloading operations, *Transportation Science*, Vol. 40, No. 4, 473-483, [doi:10.1287/trsc.1060.0148](https://doi.org/10.1287/trsc.1060.0148)
- [10] Meisel, F.; Wichmann, M. (2010). Container sequencing for quay cranes with internal reshuffles, *OR Spectrum*, Vol. 32, No. 3, 569-591, [doi:10.1007/s00291-009-0191-6](https://doi.org/10.1007/s00291-009-0191-6)
- [11] Choi, H.-S.; Kim, H.-W.; Lee, D.-H.; Yoon, J.; Yun, C. Y.; Chae, K. B. (2009). Scheduling algorithms for two-stage reentrant hybrid flow shops: minimizing makespan under the maximum allowable due dates, *International Journal of Advanced Manufacturing Technology*, Vol. 42, No. 9-10, 963-973, [doi:10.1007/s00170-008-1656-5](https://doi.org/10.1007/s00170-008-1656-5)
- [12] Atighehchian, A.; Bijari, M.; Tarkesh, H. (2009). A novel hybrid algorithm for scheduling steel-making continuous casting production, *Computers & Operations Research*, Vol. 36, No. 8, 2450-2461, [doi:10.1016/j.cor.2008.10.010](https://doi.org/10.1016/j.cor.2008.10.010)
- [13] Montoya-Torres, J. R.; Vargas-Nieto, F. (2011). Solving a bi-criteria hybrid flowshop scheduling problem occurring in apparel manufacturing, *International Journal of Information Systems and Supply Chain Management*, Vol. 4, No. 2, 42-60, [doi:10.4018/jisscm.2011040103](https://doi.org/10.4018/jisscm.2011040103)
- [14] Ruiz, R.; Vazquez-Rodriguez, J. A. (2010). The hybrid flow shop scheduling problem, *European Journal of Operational Research*, Vol. 205, No. 1, 1-18, [doi:10.1016/j.ejor.2009.09.024](https://doi.org/10.1016/j.ejor.2009.09.024)
- [15] Ribas, I.; Leisten, R.; Framinan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective, *Computers & Operations Research*, Vol. 37, No. 8, 1439-1454, [doi:10.1016/j.cor.2009.11.001](https://doi.org/10.1016/j.cor.2009.11.001)
- [16] Briskorn, D.; Drexler, A.; Hartmann, S. (2006). Inventory-based dispatching of automated guided vehicles on container terminals, *OR Spectrum*, Vol. 28, No. 4, 611-630, [doi:10.1007/s00291-006-0033-8](https://doi.org/10.1007/s00291-006-0033-8)
- [17] Santos, D. L.; Hunsucker, J. L.; Deal, D. E. (1995). Global lower bounds for flow shops with multiple processors, *European Journal of Operational Research*, Vol. 80, No. 1, 112-120, [doi:10.1016/0377-2217\(93\)E0326-S](https://doi.org/10.1016/0377-2217(93)E0326-S)
- [18] Zhang, Y.; Liang, X. L.; Li, W. F.; Zhang, Y. W. (2013). Hybrid flow shop problem with blocking and multi-product families in a maritime terminal, *10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 59-64