

Article

Joint Scheduling of Yard Crane, Yard Truck, and Quay Crane for Container Terminal Considering Vessel Stowage Plan: An Integrated Simulation-Based Optimization Approach

Hsien-Pin Hsu ^{1,*}, Chia-Nan Wang ^{2,*} , Hsin-Pin Fu ^{3,*}  and Thanh-Tuan Dang ^{2,4,*} 

- ¹ Department of Supply Chain Management, National Kaohsiung University of Science and Technology, Kaohsiung 81157, Taiwan
- ² Department of Industrial Engineering and Management, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan
- ³ Department of Marketing and Distribution Management, National Kaohsiung University of Science and Technology, Kaohsiung 824, Taiwan
- ⁴ Department of Logistics and Supply Chain Management, Hong Bang International University, Ho Chi Minh 723000, Vietnam
- * Correspondence: hphsu@nkust.edu.tw (H.-P.H.); cn.wang@nkust.edu.tw (C.-N.W.); hpfu@nkust.edu.tw (H.-P.F.); i108143106@nkust.edu.tw (T.-T.D.)

Abstract: The joint scheduling of quay cranes (QCs), yard cranes (YCs), and yard trucks (YTs) is critical to achieving good overall performance for a container terminal. However, there are only a few such integrated studies. Especially, those who have taken the vessel stowage plan (VSP) into consideration are very rare. The VSP is a plan assigning each container a stowage position in a vessel. It affects the QC operations directly and considerably. Neglecting this plan will cause problems when loading/unloading containers into/from a ship or even congest the YT and YC operations in the upstream. In this research, a framework of simulation-based optimization methods have been proposed firstly. Then, four kinds of heuristics/metaheuristics has been employed in this framework, such as sort-by-bay (SBB), genetic algorithm (GA), particle swarm optimization (PSO), and multiple groups particle swarm optimization (MGPSO), to deal with the yard crane scheduling problem (YCSP), yard truck scheduling problem (YTSP), and quay crane scheduling problem (QCSP) simultaneously for export containers, taking operational constraints into consideration. The objective aims to minimize makespan. Each of the simulation-based optimization methods includes three components, load-balancing heuristic, sequencing method, and simulation model. Experiments have been conducted to investigate the effectiveness of different simulation-based optimization methods. The results show that the MGPSO outperforms the others.

Keywords: yard crane; yard truck; quay crane; container terminal; simulation-based optimization method; experiment; statistically test



Citation: Hsu, H.-P.; Wang, C.-N.; Fu, H.-P.; Dang, T.-T. Joint Scheduling of Yard Crane, Yard Truck, and Quay Crane for Container Terminal Considering Vessel Stowage Plan: An Integrated Simulation-Based Optimization Approach. *Mathematics* **2021**, *9*, 2236. <https://doi.org/10.3390/math9182236>

Academic Editor: Frank Werner

Received: 3 August 2021

Accepted: 10 September 2021

Published: 12 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Maritime transport accounts for 80% of global trade [1], which indicates the importance of this kind of transport. One advantage of maritime transport is lower unit transportation cost, which drives the sustainability of maritime transport. Specifically, maritime transport consists of various kinds of transports, such as container transport, bulk transport, and tanker transport, among which container transport is the most popular one due to the generality and operational efficiency. Further, it is noted that about 60% of the world's deep-sea general cargos are transported through container transport, and the routes between some countries are even containerized up to 90% [2]. Thus, maritime transport has attracted our attention.

Maritime transport helps the development of international trading. This transport is especially important for ocean countries. In the world maritime transport system, port

container terminals play a major role as they not only provide basic logistics service for ships and containers but also serve as hubs for intermodal transport. However, as port container terminals are competing with each other, improving their competitiveness is essential to survive in the world maritime industry. It is found that good coordination and utilization of resources in a container terminal is an effective way to improve the operational efficiency of a container terminal.

Regarding container terminal operations, they fall into the three main areas: seaside, yard, and landside [3,4]. Each of the three areas depends on specific material handling equipment (MHE) for providing their service. The seaside area can accommodate ships and provides loading and unloading services for container ships. Berths and quay cranes (QCs) are the main resources used in this area. The berths are used to accommodate container ships, while the QCs are used for loading and unloading containers. This area includes some specific operational problems, such as the berth allocation problem (BAP), quay crane assignment problem (QCAP), and quay crane scheduling problem (QCSP) [5–7]. The landside is an area controlling the transportation of containers into/out of the container terminal. In this area, inland trucks (ITs) and trains are the commonly used MHE. Between the seaside and landside, the yard side serves as a buffer providing temporary storage for containers to be further processed for the seaside or landside. The yard cranes (YCs) and yard trucks (YTs) are the main MHE used in this area. The YCs are the MHE handling containers in this area, while YTs are used for transportation. The operational problems on the yard side include the YC assignment problem (YCAP) and YC scheduling problem (YCSP), YT assignment problem (YTAP), YT scheduling problem (YTSP), YT routing, and traffic control problem. There are some other operational problems in the yard side area, such as storage planning, space planning problem, and pre-reshuffling problem [8]. However, it is impossible to solve all these problems in one study; thus, only the YCSP, YTSP, and QCSP are considered in this research.

In [9], the authors have reviewed some studies focusing on the QCs and/or YCs. This study helps us identify the following shortcomings in the existing literature. First, many past studies focused on one single problem. However, such studies can only achieve local improvement. This neglects the fact that different MHEs can affect each other. For example, the YC can affect the YT and eventually the QC. Any bottleneck formed in the operational chain can affect the overall performance of a container terminal; thus, it should take relevant MHEs into account. Second, it is noted that the vessel stowage plan (VSP), a plan assigning vessel stowage positions to containers, has often been neglected in past research. This plan is important as it can affect the loading and unloading sequences of containers. The lack of considering this plan can cause problems in the QC operation. It may even congest the YT and YC operations on the upstream side. Taking the VSP into consideration is essential when dealing with the QCSP, YTSP, and YCSP simultaneously. Though there are two past studies, [10,11], which have considered the VSP, still this plan has been mostly neglected. The third shortcoming found in past research is that simulation-based methods have been seldom used, though this kind of approach appears suitable for dealing with complicated system problems [9], such as container terminal problems [9].

The survey in [9] shows that exact methods, evolutionary heuristics, heuristics, and simulation have been used to solve container terminal problems [9]. Mathematical models, such as mixed-integer linear programming (MILP), mixed-integer programming (MIP), constraint programming (CP), Lagrangian relaxation (LR), dynamic programming (DP), branch and bound (B&B), column generation (CG), etc., have been commonly used. However, many past studies showed the difficulty of using these exact approaches to solve a big problem to optimality due to NP-hard. Alternative approaches are still required to deal with a big instance. Most past research has thus proposed using heuristics or metaheuristics, especially the latter one. Evolutionary-based metaheuristics, such as local search-based algorithms, greedy algorithms, genetic algorithm (GA), and particle swarm optimization (PSO), have been widely used. According to [9], the simple heuristics (RULEs) and GAs top the list. However, it is found that the simulation methods have been seldom

used, though they are considered suitable for solving complicated system problems [9]. Thus, the simulation-based optimization methods have attracted our attention and are to be adopted in this research.

This research focuses on solving the simultaneous YCSP, YTSP, and QCSP by using simulation-based optimization approaches. A joint schedule for moving export containers from their current YSP positions to their VSP positions is developed with the YSP and VSP constraints taken into consideration. The objective is to minimize makespan. In the simulation-based approaches, heuristics/metaheuristics, including Method1 (sort-by-bay (SBB)), Method2 (genetic algorithm (GA)), Method3 (particle swarm optimization (PSO)), and Method4 (multiple groups particle swarm optimization (MGPSO)) have been used as sequencing methods. A small-sized experiment is used to demonstrate the solution found by Method4 (MGPSO). In addition, extensive experiments have been conducted to investigate the effectiveness of these developed approaches. Finally, a statistical *t*-test has been conducted to validate the robustness of the experimental results. The results showed that Method4 (MGPSO) outperforms the others.

The rest of this paper is organized as follows. Section 2 introduces background knowledge and relevant studies. Section 3 defines the problems that are formulated into a MILP. Section 4 proposes a framework for developing simulation-based optimization methods. Section 5 includes some experiments and analyses of the experimental results. Finally, Section 6 includes a conclusion and suggestions for future research directions.

2. Background and Relevant Literature Review

2.1. Overall View of Operations in a Container Terminal

Figure 1 shows operations in the three main areas of a container terminal. The seaside area includes facilities such as berths for accommodating ships and QCs for loading and unloading containers. The yard side area stores containers before their transportation to the seaside or yard side areas for further processing. It is a buffer between the two areas. In this area, YCs are the main MHE for storing/retrieving containers into/from a block. YTs are the main transportation means used between the seaside and yard side areas. The landside area connects to the inland and controls the transportation of containers into and out of the container terminal. The main MHE used in the landside area includes inland trucks (ITs) and trains.

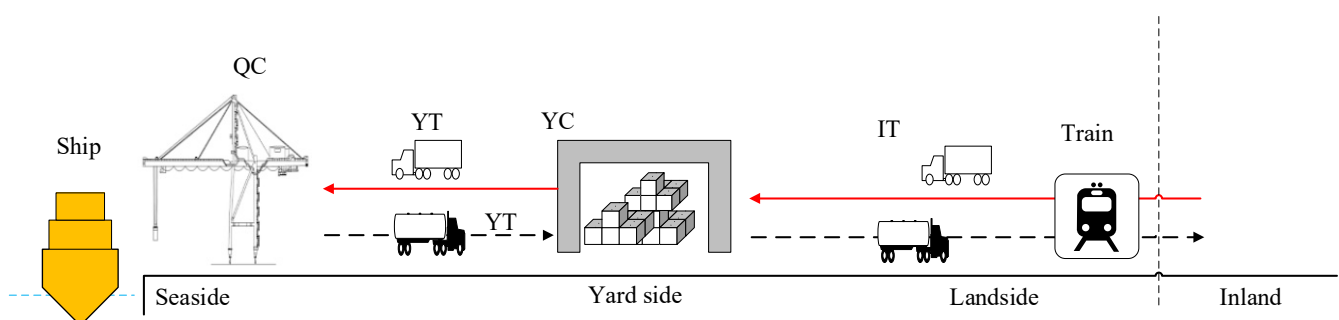


Figure 1. Operations in a container terminal.

A container terminal usually has to deal with three kinds of containers: import, export, and transshipment. These containers are moved among the seaside, yard, and landside areas. Planning storage positions for containers in different areas is an essential task for container terminal planners, and the changing of the positions of containers requires collaboration and cooperation among QCs, YTs, and YCs. A different operational procedure is used for different kinds of containers. For import containers, they are unloaded by QCs from ship firstly, transported by YTs subsequently, and stored in blocks by YCs. The export containers use the reverse procedure. Finally, these import containers are transported by ITs or trains in the landside area to their consignees. The transshipment containers use

half of the import procedure and half of the export procedure. The YCs, YTs, and QCs are essential MHE in a container terminal, and they can affect the overall performance of the container terminal considerably. To best utilize these MHEs, a joint schedule for these MHEs is necessary.

2.2. Yard Storage Plan

A yard storage plan (YSP) is a plan arranging the storage positions for containers in a block. A container yard usually consists of multiple blocks; each being configured in the Asia or Europe type [8]. Figure 2 shows a storage block of Asia type with one YC equipped. Containers are stacked along the X, Y, and Z dimensions in the block, and each container's storage position is denoted as (x, y, z), where the x, y, and z are bay, row, and tier numbers, respectively. The x, y, and z are subject to the constraint Equation (1).

$$x \leq X, y \leq Y, z \leq Z \tag{1}$$

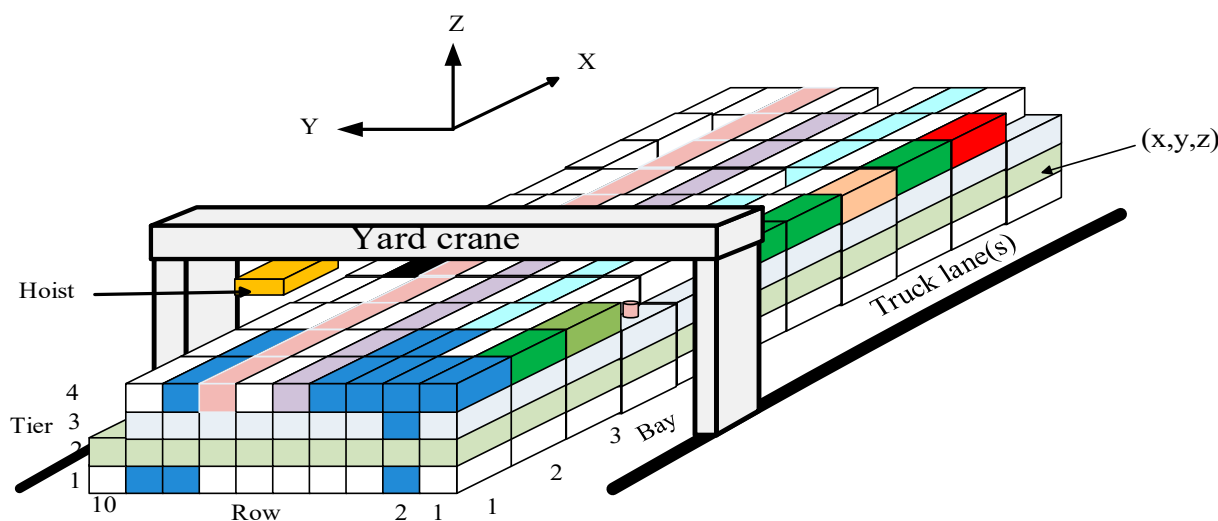


Figure 2. A storage block of Asia type with one rail mounted YC.

To access an export container, the YC and its hoist are used. The YC moves along the ±X direction to reach the bay position, and then the hoist moves along the ±Y direction to reach the row position of the target container. Both the YC and its hoist can move simultaneously. After arriving above the position of the target container, the hoist then lowers in the −Z direction to pick up the container along the +Z direction. Finally, the hoist moves the container to the YC waiting in the truck lane.

2.3. Vessel Stowage Plan

A vessel stowage plan (VSP) is a plan arranging the vessel stowage positions for export containers in a vessel. Figure 3 shows the storage positions of some bays in the vessel. The assignment of stowage positions for a container is based on the container's attributes, such as size, weight, and destination. In the vessel, each bay for storing 20-foot containers is assigned with an odd bay number, with the number increasing from the bow to the stern. In addition, two 20-foot bays can form one 40-foot bay assigned with an even bay number. For instance, together Bays 09 and 10 form Bay 11. In the VSP, each container *i* has a position denoted as (*b_i*, *r_i*, *t_i*) [10], where the *b_i* indicates a bay number; the *r_i* indicates a row number; and the *t_i* indicates a tier number.

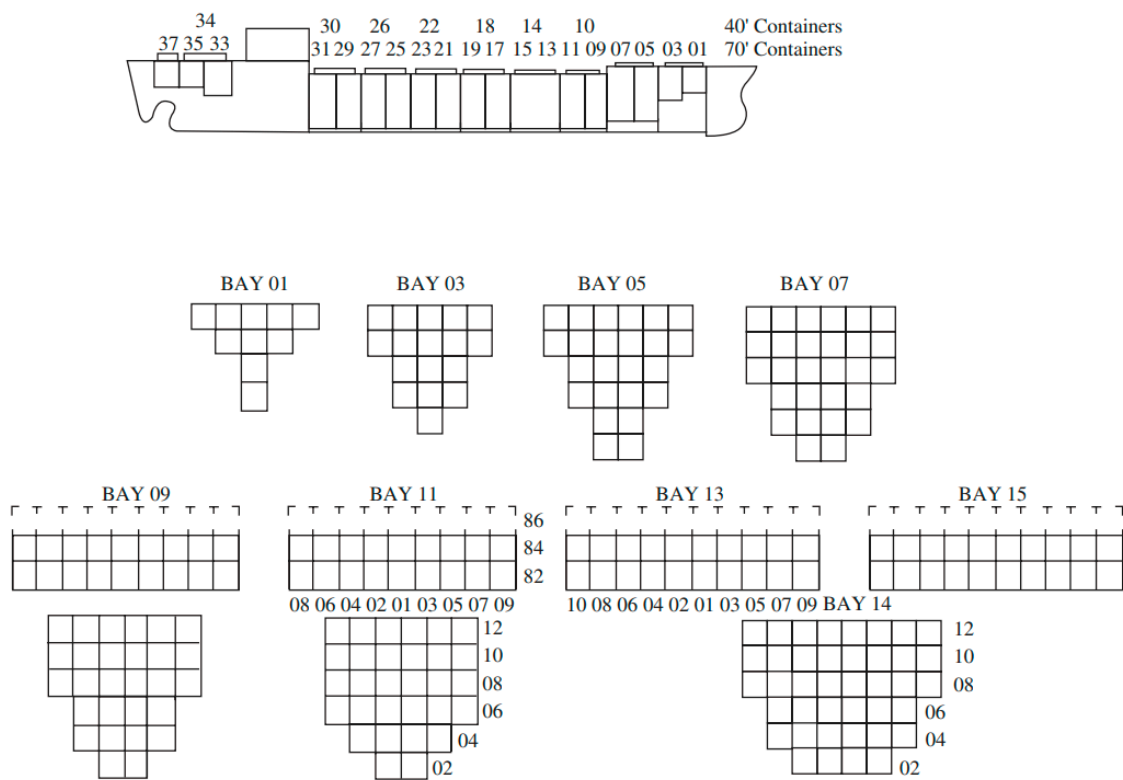


Figure 3. An example of a vessel stowage plan.

The bay structure and VSP affect the QC operations, which requires the following constraints to be considered [10].

- For some export containers to be stored at the same bay and same row, the container at a lower tier should be loaded first (i.e., a sequence constraint).
- Containers of the same bay should be assigned to the same QC (i.e., an equipment constraint).

The first one is considered as a sequence constraint, while the second one is considered as an equipment constraint. Each violation of the sequence constraints is punished with a cost defined in the cost matrix C_{ij} , as shown in Equation (2).

$$C_{ij} = \begin{bmatrix} c_{11} & \cdots & c_{1N} \\ \vdots & \ddots & \vdots \\ c_{N1} & \cdots & c_{NN} \end{bmatrix} \quad (2)$$

The $c_{ij} > 0$ is set in case container i precedes container j , which violates the constraint; Otherwise $c_{ij} = 0$. The penalty costs are considered in the objective function to rule out the solutions violating the VSP constraints seriously.

2.4. Relevant Studies

Many studies have been devoted to dealing with various container terminal problems, including QCSP [12–15], YTSP [16–19], and YCSP [20–30]. However, individual studies tend to achieve local optima due to the lack of considering the interrelationship of relevant problems.

Table 1 lists some integrated studies of some container terminal problems. Chen et al. [31] studied the QC, YT, and YC simultaneously. The authors treated the three problems as a hybrid flow shop scheduling problem (HFSP). Good coordination of these MHE was considered a key factor for the minimization of service times for ships. A mathematical model was formulated for solving these problems. However, the math-

ematical model becomes computationally intractable due to NP-hard. As an alternative, a Tabu search algorithm was another proposed to deal with a problem of big size. Zheng et al. [11] solved the QCSP together with the YCSP simultaneously, considering the YSP and VSP. The YSP assigns the storage locations for import containers, while the VSP assigns the storage position for export containers. However, this study did not consider YT. Cao et al. [32] dealt with the YCSP and YTSP simultaneously. The authors formulated a mixed-integer programming (MIP) model. However, due to computational intractability, they used Benders' decomposition method to find a solution effectively, and experiments have been conducted to investigate the effectiveness. Chen et al. [33] dealt with the QCSP, YCSP, and YTSP simultaneously by using a constraint programming (CP) model. This research considered multiple vessels with YTs shared among these vessels. The objective was to minimize the empty travel times of the QC, YC, and YT. However, this study did not take non-crossing constraints and safety margins into consideration. In addition, the CP model was found hard to solve even in a small-sized problem; thus, the authors proposed a three-stage approach, based on heuristics and disjunctive graphs, as an alternative means. This approach was found to be able to deal with a problem of up to 500 containers. Wu et al. [34] proposed a model for the integrated problem for YC and AGV in a container terminal with YSP and export container taking into consideration. The authors regarded loading as a short-term planning problem while YSP as a long-term planning problem. Furthermore, the processing sequences of containers on QCs are assumed known. A MIP model was formulated to minimize berthing times for ships. A GA was proposed to solve a big instance. Xue et al. [35] studied the integration of yard storage allocation, QCSP, and YTSP. The authors formulated a MIP for this integrated problem, which includes two weighted objectives. The first objective aims to minimize makespan, and the second objective aims to minimize the total YT travel distance. Though this study considered sequence constraints for QCs, it did not consider the non-crossing and safety distance constraints. The yard storage assignment considered block assignments instead of container slots. A two-stage heuristic algorithm was proposed to counter the computational complexity. The first stage uses an ant colony optimization (ACO) to allocate yard storage, while the second stage uses a greedy algorithm and a local search algorithm to deal with the integrated YCSP and QCSP. The authors claimed that exact approaches are impractical in solving the integrated problem. However, the generation of a lower bound remains possible. He et al. [36] studied the QCSP, YTSP, and YCSP simultaneously. The authors formulated a MIP for these problems, intending to minimize the total departure delay for all ships as well as the overall energy consumption for all tasks. An integrated simulation-based optimization approach, which includes GA and PSO, was proposed, where the GA performs the global search while the PSO performs the local search. This study illustrated the optimal trade-off between time-saving and energy-saving. Luo et al. [37] considered the integrated problem of YTSP and container storage problem. The import containers at an automated container terminal are considered. The authors formulated a MIP model for solving problems of a small size. In addition, a GA was used to deal with a big instance. However, that study did not consider QC interference and export containers. Azenodo et al. [10] solved the QCSP and the 3D stowage planning (3D SP) problem together by proposing a framework. The two problems were considered being interrelated and combinatorial. A hybrid approach combining GA with simulation was proposed as the solution means. The numbers of 30 ports, 2 QCs, and 1500 TEUs were used to test the effectiveness of this approach. According to these studies, the addition of the QCSP results in an average increase of 45.82% in load/unload time for the 3D SP problem solution. This could help the charterer avoid having to pay the ship owner for unplanned extra usage of the vessel. Kizilay et al. [38] proposed a MIP and a CP model to optimize the assignment and scheduling of QC and YC. This study treated containers as groups, called shipments, and a shipment belongs to the same port destination and the same customer. QC and YC are assigned to handle groups of containers (shipment) consecutively to simplify the problem complexity. Containers of the same shipment are

stored in the same vessel bay and storage block. It was found that the CP model was more efficient than the MIP model. Yang et al. [39] studied integrated scheduling of QCs, AGVs, and YCs, as well as the routing problems of AGVs. Both import and export containers were considered. A bi-level optimization model was developed to minimize makespan. The first level considers the integrated scheduling problem, while the second level handles the AGV routing problem. A GA-based congestion-prevention rule was proposed additionally. The authors highlighted the importance of considering these problems simultaneously. Jonker et al. [40] treated the QCSP, YCSP, and AGV as a hybrid flow shop. The authors proposed a formulation for both import and export containers. In addition, this study considered job pairs, which means that a crane can handle two containers at the same time. SA is the main method used in this research. Yang et al. [41] considered the integrated scheduling problem of the QCs, ALVs, and YCs together with the storage space assignment. The objective was to improve the overall handling efficiency and accelerate shipment. This study assumed that the sequence of containers on the QCs and the yard storage locations were known. The goal was to determine the storage site for outgoing containers and to assign tasks to the various departments involved in the ALVs and YCs, such that the makespan and energy consumption is minimized. However, the assumption of a known sequence for QCs appears to be not practical. The present study is different from the above studies. Given YSP and VSP, the YCSP, YTSP, and QCSP are to be solved simultaneously based on a load-balancing concept and by using simulation-based methods.

Table 1. The comparison of some integrated studies of container terminal problems.

Research	Problem Scopes					Method Approaches
	YSP	YC	YT	QC	VSP	
Chen et al. [31]		v	v	v		TS
Zheng et al. [11]	v	v		v	v	RULE
Cao et al. [32]		v	v			GA, RULE
Chen et al. [33]		v	v	v		CP, RULE, DG
Wu et al. [34]	v	v	v			MIP, NLMIP, GA
Xue et al. [35]	v		v	v		ACO
He et al. [36]		v	v	v		MIP, SIM, GA, PSO
Luo et al. [37]	v		v	v		MIP, GA
Azevedo et al. [10]				v	v	SIM
Kizilay et al. [38]		v		v		MIP, CP
Yang et al. [39]		v	v	v		GA
Jonker et al. [40]		v	v	v		SA
Yang et al. [41]	v	v	v	v		MIP, GA
This research	v	v	v	v	v	MILP, SIM, GA, PSO, MGPSO, SBB

Note: YSP: yard storage plan; YC: yard crane; YT: yard truck; QC: quay crane; VSP: vessel stowage plan; TS: Tabu search; RULE: rule-based heuristic; GA: genetic algorithm; CP: constraint programming; DG: disjunctive graph; MIP: mixed-integer programming; NLMIP: non-linear mixed-integer programming; ACO: ant colony optimization; SIM: simulation; PSO: particle swarm optimization; SA: simulated annealing; MGPSO: multiple groups particle swarm optimization; SBB: sort-by-bay.

3. Problem Definition, Operational Analysis, and Mathematical Model Formulation

3.1. Problem Definition

The container assignment problem, container scheduling problem, as well as VSP-oriented QCSP, YTSP, and YCSP are formally defined as follows.

Definition 1. (Container assignment problem) Let $T = \{1, \dots, |T|\}$ (the $|T|$ is the total number of tasks or containers) be a set of containers and $R = \{1, \dots, |R|\}$ (the $|R|$ is the total number of resources) be a set of resources, the container assignment is a problem of assigning a resource r ($r \in R$) to handle container j ($j \in T$), denoted as (j,r) .

Definition 2. (Container scheduling problem) Let $T = \{1, \dots, |T|\}$ be a set of containers and $R = \{1, \dots, |R|\}$ be a set of resources, the container scheduling problem is a container assignment problem (j,r) , with the resource usage duration time being also specified and denoted

as $[S_{j,r}^{RT}, E_{j,r}^{RT}]$, where the $S_{j,r}^{RT}$ and $E_{j,r}^{RT}$ are start and end times of using the resource instance r for container j , respectively. The container scheduling problem contains the container assignment problem implicitly.

Definition 3. (VSP-oriented QCSP, YTSP, and YCSP) Let $[T = \{1, \dots, |T|}]$ be a set of export containers, each container j is with the current position (x_j, y_j, z_j) in the storage block and a planned position (b_j, r_j, t_j) in a vessel, $[Q = \{1, \dots, |Q|}]$ be a set of QCs, $[K = \{1, \dots, |K|}]$ be a set of YTs, and $[P = \{1, \dots, |P|}]$ be a set of YCs, the VSP-oriented QCSP, YTSP, and YCSP is an integrated scheduling problem arranging the movements of each container j from the position (x_j, y_j, z_j) to the (b_j, r_j, t_j) by using YC, YT, and QC and considering the VSP constraints. Each usage duration of resource r is denoted as $[S_{j,r}^{RT}, E_{j,r}^{RT}]$, where the $S_{j,r}^{RT}$ and $E_{j,r}^{RT}$ are start and end usage times, respectively. The $r \in R_{RT} = P \cup K \cup Q$. The RT indicates a resource type. If $RT = 1$ then $r \in P$; if $RT = 2$ then $r \in K$; if $RT = 3$ then $r \in Q$. Minimizing makespan (completion time) is the objective when fulfilling the VSP with sequence constraints taking into consideration.

To find solutions to the simultaneous YCSP, YTSP and QCSP, methodologies, such as mathematical model, heuristic, metaheuristic, and simulation, are available.

3.2. Operations Analysis

3.2.1. Yard Crane (YC) Operation Analysis

Figure 4 shows the top view of a storage block that is equipped with two YCs. The operations of the hoist on YC1 are analyzed as follows.

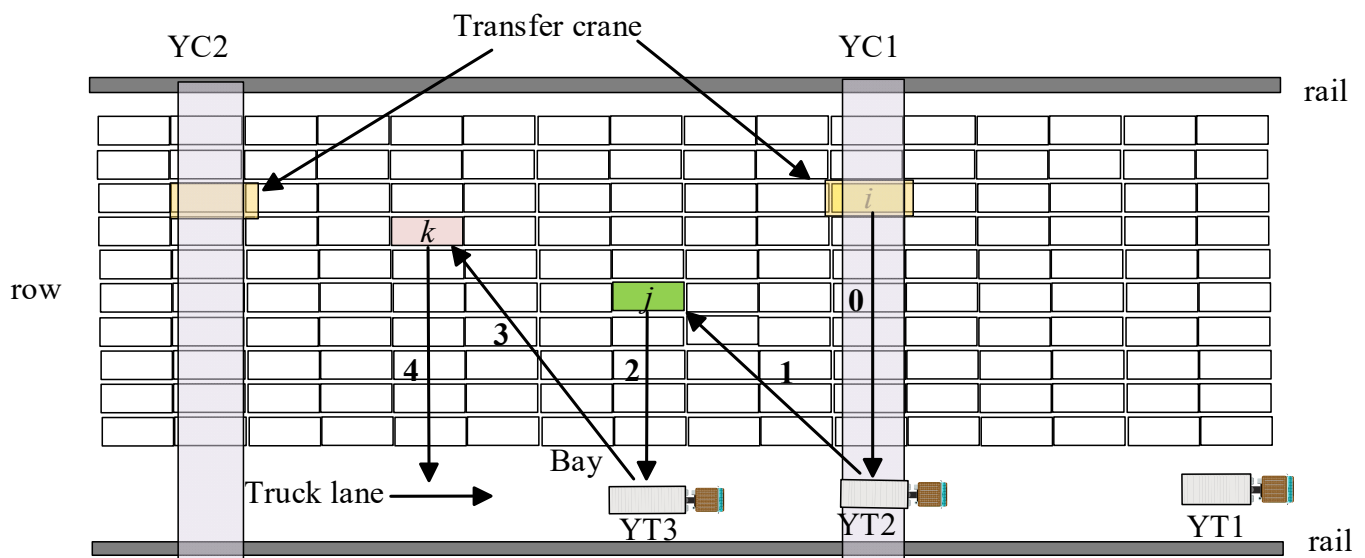


Figure 4. The movement of the hoist on the yard crane.

Let containers i, j , and k be in sequence to be picked up by this hoist, then the routing paths 0,1,2,3, and 4 will be used sequentially. After loading container i onto the YT 2 along path 0, the hoist will move to container j along path 1. After picking up container j , the hoist will move container j to the truck lane along path 2 and finally load it onto a YT. The YSP provides position information of containers in the block. Given (x_i, y_i, z_i) and (x_j, y_j, z_j) as the storage positions of containers i and j , then Equation (3) is used to estimate the total time required for the hoist to load container j onto a YT.

$$PT1_{ij} = \text{Max}\{d(|x_i - x_j|) / v_x, d(y_j) / v_y\} + 2 \cdot d(H - z_j) / v_z + d(y_j) / v_y + 2 \cdot d(H - 1) / v_z \tag{3}$$

where,

v_x : is the moving velocity of YC along the x -direction.

v_y : is the moving velocity of YC along the y -direction.

v_z : is the moving velocity of YC along the z -direction.

$d(\bullet)$: is a function transforming container coordinate into distance.

The right-hand side includes four-time components. The first component indicates the transportation time for the hoist to reach the bay of container j (path 1), where the $d(|x_i - x_j|)$ is the distance for moving from the x_i to x_j and the $d(y_j)$ is the distance for reaching the row y_j of container j . The second component is the roundtrip time to pick up container j , where $d(H - z_j)$ is the moving distance. The third component is the time for moving container j to the truck lane (path 2), where $d(y_j)$ is the moving distance. The fourth component is the roundtrip time to load container j onto a YT, where $d(H - 1)$ is the moving distance.

Given that $S_{j,r}^1$ is the start time to process container j , then Equation (4) is used to estimate the completion time.

$$E_{j,r}^1 = S_{j,r}^1 + PT1_{ij} \tag{4}$$

3.2.2. Yard Truck (YT) Operation Analysis

YTs move containers between QCs and YCs. Usually, a fleet of YTs are assigned to a ship, and these YTs run in a loop, one after another. The roundtrip time ($PT2_j$) can be assumed as a fixed time. Minimizing the waiting time for YTs is important.

Let $\min_{j \in T} \{a_j\}$ be the available time of the earliest available container j (after YC operation) and $\min_{k \in K} \{a_k\}$ is the available time of the earliest available YT k , then Equation (5) is the earliest available start time for the YT k to transport the container.

$$S_{j,k}^2 = \text{Max}\left\{ \text{Min}_{j \in T} \{a_j\}, \text{Min}_{k \in K} \{a_k\} \right\} \tag{5}$$

This can minimize the waiting times for both container and YT. Given $PT2_j$ as the roundtrip time, then Equation (6) is used to estimate the end transportation time for container j .

$$E_{j,k}^2 = S_{j,k}^2 + PT2_j = \text{Max}\left\{ \text{Min}_{j \in T} \{a_j\}, \text{Min}_{k \in K} \{a_k\} \right\} + PT2_j \tag{6}$$

3.2.3. Quay Crane (QC) Operation Analysis

Figure 5 shows that two QCs are working for a ship. To solve the QCSP, it needs to consider the following constraints.

- Each QC handles one first task and one last task.
- Each task can only be handled by one QC.
- Each QC handles a sequence of containers based on their positions given by a VSP.
- The interference of QCs should be avoided.
- Given a set of containers to be loaded into a ship at the same bay and column, the container with the lowest tier should be loaded firstly.

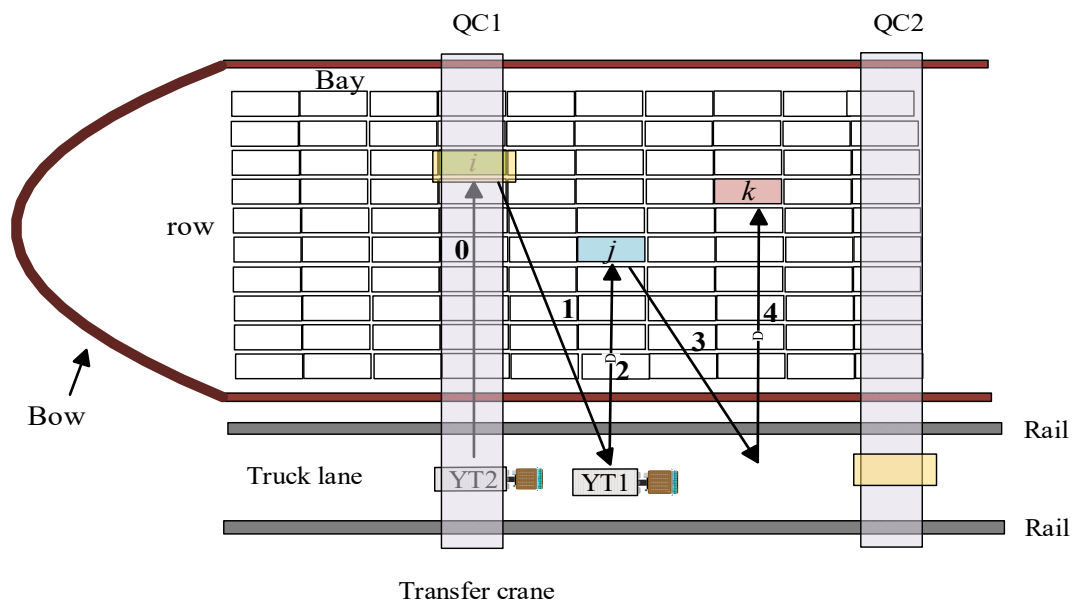


Figure 5. The YT and QC operations on the seaside.

Assume that containers i, j , and k are in a sequence to be loaded by the hoist on QC 1 into the ship. Then, the routing paths 0,1,2,3, and 4 will be used sequentially. After loading container i , the hoist will move along path 1 to access container j . The VSP provides position information of containers in the vessel. Given that (b_i, r_i, t_i) and (b_j, r_j, t_j) are the vessel storage positions of containers i and j , then the loading time for container j is estimated by Equation (7).

$$PT3_{ij} = \text{Max}\{d(|b_i - b_j|) / v_b, d(r_i) / v_r\} \cdot 2 \cdot d(\hat{H}) / v_t + d(r_j) / v_r + 2 \cdot d(t_j) / v_t \quad (7)$$

where,

v_b : is the moving velocity of QC along the *bay* direction.

v_r : is the moving velocity of QC along the *row* direction.

v_t : is the moving velocity of QC along the *tier* direction.

\hat{H} : is the height for a QC to pick up a container from a YT.

$d(\bullet)$: is a function transforming container coordinate into distance.

The right-hand side also includes four time components. The first component is the time for the hoist to reach truck lane of bay b_j , where the $d(|b_i - b_j|)$ is the moving distance to reach the bay b_j , and the $d(r_i)$ is the moving distance to reach the truck lane. The second component is the roundtrip time for picking up container j from the YT 1. The third component is the moving time to reach the row position r_j , where the $d(r_j)$ is the moving distance. The fourth component is the roundtrip time to load container j into the vessel at its tier position t_j , where the $d(t_j)$ is the moving distance.

Given a_j as the available time of container j ($j \in T$) and a_q as the available time of the assigned QC q ($q \in Q$), Equations (8) and (9) are used to determine the start and end usage times of the QC q , respectively.

$$S_{j,q}^3 = \text{Max}\{a_j, a_q\} \quad (8)$$

$$E_{j,q}^3 = S_{j,q}^3 + PT3_{i,j} = \text{Max}\{a_j, a_q\} + PT3_{ij} \quad (9)$$

3.3. Mathematical Model

This sub-section formulates a mathematical model for the YCSP, YTSP, and QCSP simultaneously. The assumptions, indices, sets, parameters (input data), and decision variables are first introduced as follows.

Assumptions

- All export containers have a storage position (x,y,z) in a block and a stowage position (b,r,t) in a vessel.
- YCs and QCs are homogeneous in moving speed.
- No-passing each other for YCs and QCs.
- No interruptions during loading and transporting containers.
- No relocation of containers.

Indices

i, j	a container number; $i, j \in T = \{1, \dots, T \}$.
p	a YC number; $p \in P = \{1, \dots, P \}$.
k	a YT number; $k \in K = \{1, \dots, K \}$.
q	a QC number; $q \in Q = \{1, \dots, Q \}$.

Set

T	a set of containers (tasks) $T = \{1, \dots, T \}$.
RT	a set of resource types; $RT = \{1, 2, 3\}$. $RT = 1$ indicates YC; $RT = 2$ indicates YT; $RT = 3$ indicates QC.
B	a set of vessel bays $B = \{1, \dots, B \}$.
K	a set of yard trucks $K = \{1, \dots, K \}$.
P	a set of yard cranes $P = \{1, \dots, P \}$.
Q	a set of quay cranes $Q = \{1, \dots, Q \}$.

Parameters (Input data)

$ T $	total number of tasks (containers).
(X, Y, Z)	dimensions of a storage block. X, Y, Z are the maximum number of bays, rows, tiers, respectively.
(x_j, y_j, z_j)	the coordinate of a container j in a block.
(b_j, r_j, t_j)	the coordinate of a container j in a vessel.
$ P $	total number of YCs.
$ K $	total number of YTs.
$ Q $	total number of QCs.
$ B $	total number of bays in the ship.
\hat{H}	the height for a QC to pick up a container from a YT.
H	the tier used by hoist to move containers across the storage block ($H = Z + 1$).
v_x	moving speed of YC along the X -direction.
v_y	moving speed of YC along the Y -direction.
v_z	moving speed of YC along the Z -direction.
v_b	moving speed of QC along the <i>bay</i> direction.
v_r	moving speed of QC along the <i>row</i> direction.
v_t	moving speed of QC along the <i>tier</i> direction.
a_i	available time of container i .
a_p	available time of YC p .
a_k	available time of YT k .
a_q	available time of QC q .
$PT2_j$	travel time for YT transporting container j to QC (a fixed time).
M	a big number.

Decision variables

X_{ijp}	= 1, if container i is loaded before j by YC q ; = 0, otherwise.
Y_{ijk}	= 1, if container i is transported before j by YT k ; = 0, otherwise.
Z_{ijq}	= 1, if container i is loaded before j by QC q ; = 0, otherwise.
N_p	total number of containers assigned to YC p .
N_k	total number of containers assigned to YT k .
N_q	total number of containers assigned to QC q .
$PT1_{ij}$	total YC loading time of container j , estimated by Equation (3).
$PT3_{ij}$	total QC loading time of container j , estimated by Equation (7).
$S_{j,r}^{RT}$	start usage time of resource instance r of the type RT for container j ($j \in T$). if $RT = 1$ then $r \in P$; if $RT = 2$ then $r \in K$; if $RT = 3$ then $r \in Q$.
$E_{j,r}^{RT}$	end usage time of resource instance r of the type RT for container j ($j \in T$). if $RT = 1$ then $r \in P$; if $RT = 2$ then $r \in K$; if $RT = 3$ then $r \in Q$.

The mixed-integer linear programming (MILP) model of the simultaneous YCSP, YTSP, and YQSP is formulated as follows.

$$\text{Min } Z = \text{Max } (E_{j,q}^3) \tag{10}$$

such that

$$\sum_{i \in T} \sum_{j \in T} \sum_{p \in P} X_{ijp} = ||T|| \tag{11}$$

$$\sum_{j \in T} X_{ijp} = 1, \forall i \in T, \forall p \in P \tag{12}$$

$$\sum_{i \in T} X_{ijp} = 1, \forall j \in T, \forall p \in P \tag{13}$$

$$\sum_{i \in T} \sum_{j \in T} X_{ijp} = N_p \leq N, \forall p \in P \tag{14}$$

$$\sum_{i \in T} \sum_{j \in T} \sum_{k \in T} Y_{ijk} = ||T|| \tag{15}$$

$$\sum_{j \in T} Y_{ijk} = 1, \forall i \in T, \forall k \in K \tag{16}$$

$$\sum_{i \in T} Y_{ijk} = 1, \forall j \in T, \forall k \in K \tag{17}$$

$$\sum_{i \in T} \sum_{j \in T} Y_{ijk} = N_k \leq N, \forall k \in K \tag{18}$$

$$\sum_{i \in T} \sum_{j \in T} \sum_{q \in Q} Z_{ijq} = ||T|| \tag{19}$$

$$\sum_{j \in T} Z_{ijq} = 1, \forall i \in T, \forall q \in Q \tag{20}$$

$$\sum_{i \in T} Z_{ijq} = 1, \forall j \in T, \forall q \in Q \tag{21}$$

$$\sum_{i \in T} \sum_{j \in T} Z_{ijq} = N_q \leq ||T||, \forall q \in Q \tag{22}$$

$$(S1_j - S1_i) + M(1 - X_{ijq}) \geq P1_{ij}, \forall i, j \in T, \forall j \in T, \forall q \in Q \tag{23}$$

$$(S2_j - S2_i) + M(1 - Y_{ijk}) \geq P2_i, \forall i, j \in T, \forall j \in T, \forall q \in Q \tag{24}$$

$$(S3_j - S3_i) + M(1 - Z_{ijk}) \geq P3_{ij}, \forall i, j \in T, \forall j \in T, \forall q \in Q \quad (25)$$

$$S2_j \geq E1_j, \forall j \in T \quad (26)$$

$$S3_j \geq E2_j, \forall j \in T \quad (27)$$

$$S_{j,r}^{RT}, E_{j,r}^{RT} \geq 0, \forall j \in T, \forall r \in K \cup Q \cup P, \forall RT \in R \quad (28)$$

$$X_{ijp}, Y_{ijk}, Z_{ijq} \in \{0, 1\}, \forall i, j \in T, \forall k \in K, \forall p \in P, \forall q \in Q \quad (29)$$

Equation (10) is the objective function minimizing the maximum completion times (makespan) of containers. Equations (1)–(9) are also included in this model. Constraints (11)–(14) relate to the decision variable X_{ijp} . Constraint (11) requires the total number of containers assigned to YCs is the exact number $||T||$. Constraint (12) ensures container i has only one successor assigned to the same YC. Constraint (13) ensures container j has only one predecessor assigned to the same YC. Constraint (14) finds the total number of containers assigned to a given YC p (i.e., N_p), and the N_p should not be greater than N . Constraints (15)–(18) relate to the decision variable Y_{ijk} . Constraint (15) stipulates that the total number of containers assigned to YTs is the exact number of $||T||$. Constraint (16) ensures container i has only one successor j assigned to the same YT. Constraint (17) ensures container j has only one predecessor i assigned to the same YT. Constraint (18) finds the total number of containers assigned to a given YT k (i.e., N_k), and the N_k should not be greater than $||T||$. Constraints (19)–(22) relate to the decision variable Z_{ijq} . Constraint (19) stipulates that the total number of containers assigned to YQs is the exact number of $||T||$. Constraint (20) ensures container i has only one successor j assigned to the same QC. Constraint (21) ensures that container j has only one predecessor i assigned to the same QC. Constraint (22) finds the total number of containers assigned to a given QC q (i.e., N_q), and the N_q should not be greater than $||T||$. In order to comply with constraint (23), the loading of container j (successor) must be completed before container i (predecessor) assigned to the same YC. It is a requirement of constraint (24) that the transportation of container j (successor) does not take place before the completion of container i (predecessor) assigned to the same YT. According to constraint (25), the start time of YT carrying of a container cannot be earlier than the completion time of YC loading of the same container. This constraint links the YCSP and YTSP. Constraint (26) requires the start transportation time of YT for a container cannot before its YC completion time. This constraint links the QCSP and YTSP. Constraint (27) requires the start time of QC loading of a container cannot be before the completion time of YT transportation. Constraint (28) defines that the values of decision variables $S_{j,r}^{RT}$ and $E_{j,r}^{RT} \geq 0$. Constraint (29) defines that X_{ijp} , Y_{ijk} , and Z_{ijq} are binary variables.

4. Simulation-Based Optimization Methods

Due to NP-hard, the MILP developed in the previous section will become computational intractable when dealing with a big instance. Thus, simulation-based methods are developed in this sub-section as the actual solution means.

4.1. The Framework of Simulation-Based Optimization Methods

Figure 6 shows the framework of simulation-based optimization methods for dealing with the integrated problem of YCSP, YTSP, and QCSP. It includes the following steps.

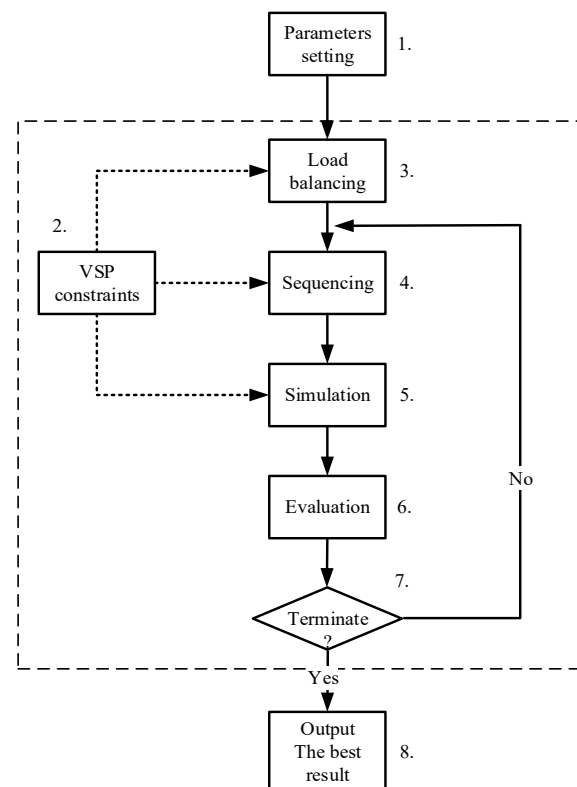


Figure 6. The framework of a simulation-based optimization approach.

Step 1: Set parameter values for a container storage block (such as $X, Y, Z, Q, H, v_x, v_y, v_z$, etc.), methods (such as $\|T\|, P, T$, etc.), and relevant data (x_j, y_j, z_j) of YSP and (b_j, r_j, t_j) of VSP. Set penalty cost PC .

Step 2: Discover the sequence constraints required for VSP based on (b_j, r_j, t_j) data. The cost matrix C_{ji} is generated automatically. Each violation is imposed as a penalty cost PC .

Step 3: Balance workload for available QCs and YCs by using the load-balancing heuristic. This workload-balancing strategy is expected to help achieve a good operational result.

Step 4: Form the loading sequence of containers for each YC by using different methods, such as PSO, MGPSO, GA, SBB, and the ROV technique. Alternative sequences of containers will be created. These sequences of containers will serve as inputs to Step 5.

Step 5: Simulate YC, YT, and QC operations by using the simulation model. The alternative loading sequences of containers determined in Step 4 will be used in this step.

Step 6: Evaluate the simulation results and store the best solution.

Step 7: Check the termination condition at the end of each iteration. If the termination condition is met, then go to step 8; otherwise, go to step 4.

Step 8: Output the best planning result.

In the simulation-based optimization methods, the load-balancing, sequencing, and simulation models are three core components. Each of them is detailed in the following sub-sections.

4.2. The Load-Balancing Heuristic

The load-balancing heuristic aims to balance the workload among equipment (QCs and YCs), considering two constraints. The first constraint requires that containers of the same bay have to be assigned to the same equipment. The second constraint stipulates that equipment cannot cross over each other. Algorithm 1 includes the logic of workload balance applicable to either QCs or YCs.

Algorithm 1 The Logic of Workload Balance for Equipment (QCs or YCs)

```

1:  Set the parameter values for ||T|| and ||Q||; set  $WL_q = 0$  ( $q = 1, \dots, Q$ ).
2:  Find the workload limit for QC/YC  $q$  by using (30).

                                      $WL = \text{INT}(\|T\|/\|Q\|)$ 

3:  Sort containers increasingly by their bay numbers into an ascending list T; set  $i = 1; q = 1$ .
4:  For ( $i = 1; i \leq \|T\|; i++$ )
5:    Assign the  $i$ -th container in T to the QC/YC  $q$ 
6:     $WL_q = WL_q + 1$ 
7:    IF ( $WL_q \geq WL$ ) // workload assessment
8:      IF ( $y_i \neq y_{i-1}$ ) // different bay numbers of two containers
9:         $q = q + 1$  // change to the next QC/YC
10:     END IF
11:  END IF
12:  END For

```

In this algorithm, Step (1) first initializes the following parameter values: $\|T\|$ (total number of containers), Q (total number of QCs), and WL_q (workload of QC $q \in Q$). Equation (30) estimates the workload limitation for each QC in terms of the number of containers. The $\text{INT}(\bullet)$ is a function rounding a real value to its nearest integer. Step (7) checks whether the workload limitation has been reached? if “Yes,” it considers the change of QC for assignment. However, Step (8) makes the final decision. Does this step check whether the two consecutive containers, at the i -th and $(i-1)$ -th orders, are in the same bay? if “No” then change the assignment to the next QC; otherwise, no change. This heuristic separates QCs into different working spaces.

4.3. The Main Flow Logic of MGPSO

Heuristics and metaheuristics, such as PSO and MGPSO (an improved PSO), are used as sequencing methods, which are detailed as follows.

4.3.1. Particle Swarm Optimization (PSO)

Proposed by Kennedy and Eberhart [42], the PSO searches a solution space by using a group of particles. Let $X_i(t)$, $V_i(t)$, and $P_i(t)$ be the position, velocity, and personal best position vector of the particle i and $X_g(t)$ be the position of the global best particle g , then the position of the particle i at the time $t+1$ is determined by Equation (31).

$$V_i(t+1) = wV_i(t) + c_1R_1(P_i(t) - X_i(t)) - c_2R_2(X_g(t) - X_i(t)) \quad (30)$$

where w is an inertia weight within the interval $(0,1)$; c_1 and c_2 are acceleration coefficients usually set to the value 2.0; R_1 and R_2 are random numbers within the interval $(0,1)$. The velocity $V_i(t)$ of a particle is limited within the range (\bar{V}, \underline{V}) , where \bar{V} is the maximum velocity while \underline{V} is the minimum velocity allowed for a particle.

Updating velocity in this way enables particles to search in the directions of their personal best and the global best. The next position of particle i is determined by Equation (32).

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (31)$$

Figure 7 shows the encoding scheme of position for particles, which includes two segments. Each segment contains N dimensions, which corresponds to the total number of containers. The load-balancing heuristic is used to find a solution for the left segment, in which each u_i indicates the YC assigned to the container i . Take the vector V to become $(1,2,2,1,2,1 | 0.2,0.4,0.1,0.5,0.3,0.9)$ as an example, it shows that the Containers 1, 4, and 6 are assigned to YC 11 while Containers 2, 3, and 5 are assigned to YC 2.

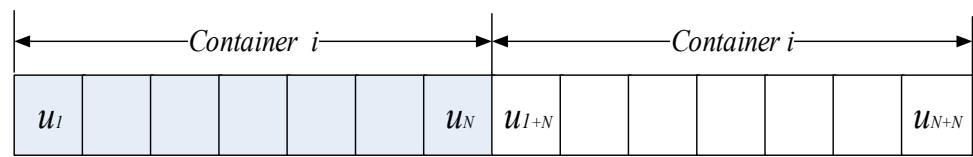


Figure 7. The encoding scheme of position.

The right segment encodes the sequences of containers to be processed by YCs. However, the ROV technique is first used to transform the real values u_{i+N} ($i = 1, \dots, N$), in increasing order, into ranking numbers, each corresponding to the loading order of container i . For instance, the right segment of the vector V is transformed into (1,3,1,2,2,3), indicating that the sequence 1-4-6 of containers is to be processed by YC 1, while the sequence 3-5-2 of containers is to be processed by YC 2.

The PSO is used as a sequencing method for the simulation-based optimization method, Method3 (PSO).

4.3.2. Multiple Groups Particle Swarm Optimization (MGPSO)

The MGPSO, which is an improved PSO, has the basic flow as shown in Figure 8. Each of the steps is detailed as follows.

- Step 1: Set parameter values.
- Step 2: Initialize particle positions.
- Step 3: Evaluate and rank particles.
- Step 4: Reshuffling particles into groups, which is performed at the beginning of each iteration.
- Step 5: Move a particle toward the group best.
- Step 6: Check whether the position of this particle is improved? if “Yes” then go to Step 12; if “No” then go to Step 7.
- Step 7: Move this particle toward the global best.
- Step 8: Check whether the position is better? if “Yes” then go to Step 9; if “No” then go to Step 10.
- Step 9: Accept the position for this particle.
- Step 10: Give a random position for this particle.
- Step 11: Accept this position.
- Step 12: Check whether to change to the next particle? if “Yes” then go to Step 5; if “No” then go to Step 13.
- Step 13: Check whether to change to the next group? if “Yes” then go to Step 5; if “No” then go to Step 14.
- Step 14: Check whether to change to the next iteration? if “Yes” then go to Step 3; if “No” then go to Step 15.
- Step 15: End.

The MGPSO is different from the PSO with the following new features: (1) multiple groups of particles, (2) reshuffling particles at the beginning of each iteration, (3) decreasing number of groups, and (4) adaptive velocity [4].

Feature (1) enables particles to be influenced by more elites, i.e., the best particles of groups. This helps diversify particles. The particles are grouped in this way. Given the number of m groups, the best particle is assigned to the 1st group; the second-best to the 2nd group; the (m)th best to the m -th group; the ($m+1$)th back to the 1st group, the ($m+2$)th back to the 2nd group, and so on. Feature (2) can intensify the diversity for particles as they can change the belonging group. Feature (3) helps implement the strategy of exploration-to-exploitation as more groups enable wide exploration in a solution space while more particles in a group intensify exploitation on the best elite in each group. Equation (33) is used to determine the group numbers at each iteration t .

$$GN(t) = \text{INT}(\sqrt{P} + \sqrt{P}/2 - t \cdot \sqrt{P}/T) \tag{32}$$

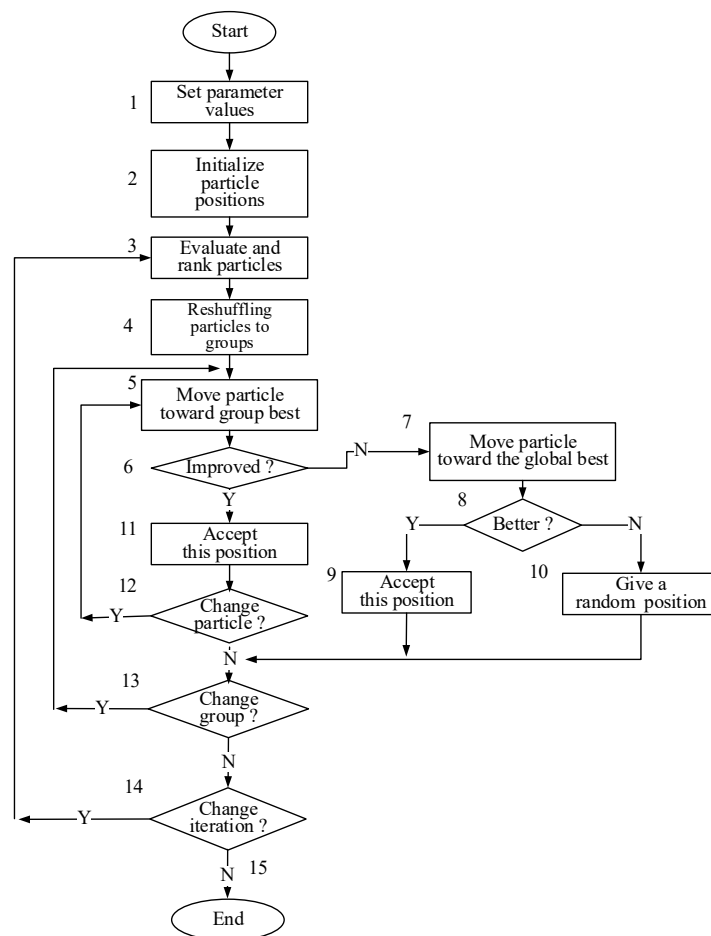


Figure 8. The basic logic flow of the MGPSO.

The function $\text{INT}(\bullet)$ rounds a real value to its nearest integer. The P is total number of particles. The T is total number of iterations. At the first iteration, $t = 1$, a minimum momentum is created to explore the solution space. At the last iteration, $t = T$, a maximum momentum is created to exploit the best elite of each group. Finally, Feature (4) enables an adaptive fly for a particle based on its current position relative to the position of a target particle (which can be the best particle of the same group or the global best particle in the swarm). Equation (34) is the adaptive velocity used by the MGPSO, which is an improvement of Equation (31) of PSO, in which the random numbers $R1$ and $R2$ tend to make random fly.

$$\tilde{V}_j(t) = D_{o,j}(t) \oplus \Delta V_j(t) \tag{33}$$

In addition, Equation (35) is used in the MGPSO to determine the next position for a particle.

$$X_j(t + 1) = X_j(t) \odot \tilde{V}_j(t), \tag{34}$$

The $D_{o,j}(t)$ is termed as *total distance vector* and is obtained by Equation (36).

$$D_{o,j}(t) = [X_{o,k}(t) \sim X_{j,k}(t); k = 1, \dots, D] \tag{35}$$

The k -th element of the $D_{o,j}(t)$ is determined by Equation (37).

$$X_{o,k}(t) \sim X_{j,k}(t) = \begin{cases} 0, & \text{if } X_{o,k}(t) = X_{j,k}(t) \\ X_{o,k}(t), & \text{if } X_{o,k}(t) \neq X_{j,k}(t) \end{cases} \tag{36}$$

The $\Delta V_j(t)$, which is a binary vector, is termed an *adaptive velocity vector*. Its k -th element is determined by Equation (38).

$$\Delta V_{j,k}(t) = \begin{cases} 0, & \text{if } R(\cdot) \geq RB1_j(t) \\ 1, & \text{if } R(\cdot) < RB1_j(t) \end{cases} \quad (37)$$

where the $R(\cdot)$ is a random number and the $RB1_j(t)$ is a rate of binary value 1, a probability used to control the generation of the binary value 1. Both $R(\cdot)$ and $RB1_j(t)$ within the interval (0,1).

The $RB1_j(t)$ is determined by Equation (39).

$$RB1_j(t) = \begin{cases} \frac{|D_{o,j}(t)|-2}{D}, & \text{if } |D_{o,j}(t)| > 2 \\ 0, & \text{if } |D_{o,j}(t)| \leq 2 \end{cases} \quad (38)$$

The $|D_{o,j}(t)|$ is termed as *Hamming distance*, which is obtained by Equation (40).

$$|D_{o,j}(t)| = D - \sum_{k=1}^D (X_{o,k}(t) \sim X_{j,k}(t)) / X_{o,k}(t) \quad (39)$$

The *Hamming distance*, in fact, counts the number of different elements between two position vectors.

In Equation (41), the operator " \oplus " is termed as *adaptive distance operator*. It works as follows.

$$D_{o,j}(t) \oplus \Delta V_{j,k}(t) = \begin{cases} D_{o,j}(t), & \text{if } \Delta V_{j,k}(t) = 1 \\ 0, & \text{if } \Delta V_{j,k}(t) = 0 \end{cases} \quad \text{for the } k\text{-th element} \quad (40)$$

In Equation (35), the operator " \odot " uses the following steps: (1) it finds and adopts the first non-zero value out of the \tilde{V}_j and replaces the value at the same position in the vector $X_j(t)$, (2) the replaced value takes the position of the non-zero value in the $X_j(t)$, (3) it repeats the Steps (1) and (2) until there is no non-zero value in the \tilde{V}_j , and (4) the modified $X_j(t)$ becomes the next position $X_j(t + 1)$.

In addition to the four features mentioned above. The MGPSO also uses *Tabu fly* and *Neighborhood search* for moving particles. The *Tabu fly* aims to stop a particle from flying to the target particle o directly. This will waste one local search as this position has been visited by the target particle. Two steps are used for the *Tabu fly*. First, it measures the *Harming distance* between particles j and o . If the *Harming distance* ≤ 2 then it stops generating the binary value 1 for the vector $\Delta V_j(t)$ to avoid a direct fly. On the other hand, the *Neighborhood search* aims to improve the *Tabu fly*, as this mechanism can keep a particle at its current position, which can also waste one local search. The *Neighborhood search* uses the neighborhood (p1,p2) function, which switches two elements of a position vector. It initiates a local search for the particle with a *Tabu fly*.

Given $P_j(t) = (1, 2, 3, 4)$ and $P_o(t) = (4, 3, 2, 1)$ as positions of particles j and o at the time t , respectively, Figure 9 illustrates the movement of the particle j to its next position to approach target particle o .

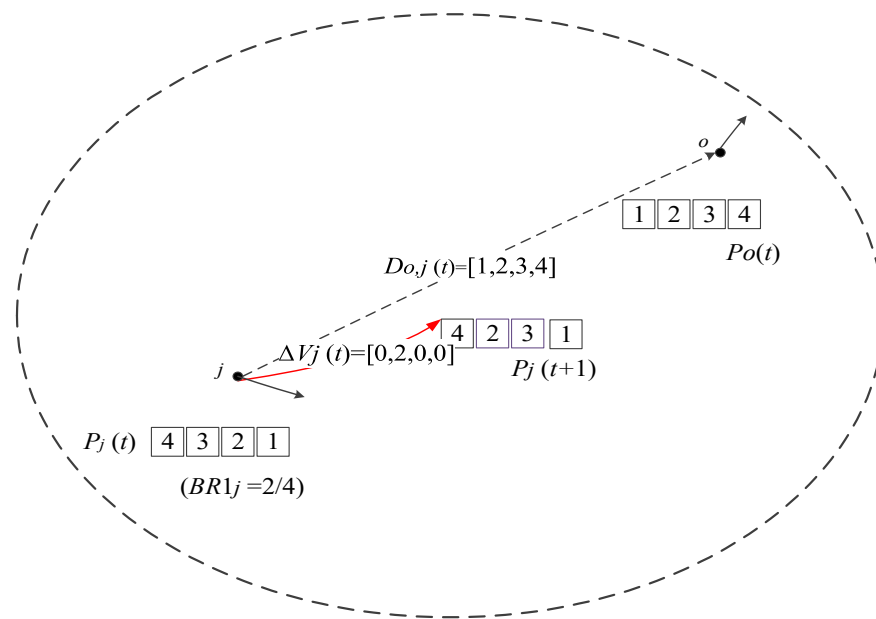


Figure 9. Particle i searches around the target particle o within a group.

The following steps are used:

1. Determine the total distance vector by using Equation (34). The total distance vector is then derived as follows.

$$D_{o,j}(t) = [1, 2, 3, 4] \sim [4, 3, 2, 1] = [1, 2, 3, 4]$$

2. Determine the $RB1_j(t)$ and $|D_{o,j}(t)|$ by using Equations (39) and (40).

$$RB1_j(t) = 2/4 \text{ and } |D_{o,j}(t)| = 4$$

3. Determine the adaptive velocity vector by using Equations (38) and (39). Here, we assume $\Delta V_j(t) = (0, 1, 0, 0)$.
4. Determine the adaptive velocity using Equation (34). We can get the $\tilde{V}_j(t)$ as follow.

$$\tilde{V}_j(t) = (1, 2, 3, 4) \oplus (0, 1, 0, 0) = (0, 2, 0, 0)$$

5. Determine the next position $X_j(t + 1)$ using Equation (35).

$$X_j(t + 1) = X_j(t) \odot \tilde{V}_j(t) = (4, 3, 2, 1) \odot (0, 2, 0, 0) = (4, 2, 3, 1)$$

4.4. The Simulation Method

The simulation is based on a simulation model termed as Timed Predicate/Transition net (TP/T net) [43,44].

Definition 4. Timed Predicate/Transition net (TP/T net) is defined as 6- tuple:

$$\text{Timed Pr/Tr net} = (P, T, A, \Sigma, L, F)$$

where

- P: a set of predicates; $P = P_t \cup P_{nt}$, where P_t is a set of timed predicates; P_{nt} is a set non-timed predicates. $P_i \in P_t$ or P_{nt} and $P_t \cap P_{nt} = \emptyset$.
- T: a set of transitions $\{T_1, \dots, T_n\}$.
- A: a set of arcs.
- Σ : a structure including individual tokens (T_i) as well as operations (O_j) and relations (R_k), i.e., $\Sigma = (T_1, \dots, T_i; O_1, \dots, O_j; R_1, \dots, R_k)$.
- L: labeling of arcs with a formal sum of n attributes of token's variables, consisting of zero-attributes indicating a no-argument token.
- F: a set of inscriptions in transitions; being logical formulas established from the operations and relations of the structure Σ ; variables occurring free in a formula have to be at an adjacent arc.
- M: a marking of P with formal sums of n -tuples of individual tokens. M_0 is an initial marking.

Figure 10 shows the TP/T net model used in this research. It includes predicates $P = P_t \cup P_{nt} = \{Using_R\} \cup \{Open_task, Close_task, Avail_R\}$, directed arcs $A = \{A1, \dots, A9\}$, transitions $T = \{T1, T2, T3\}$.

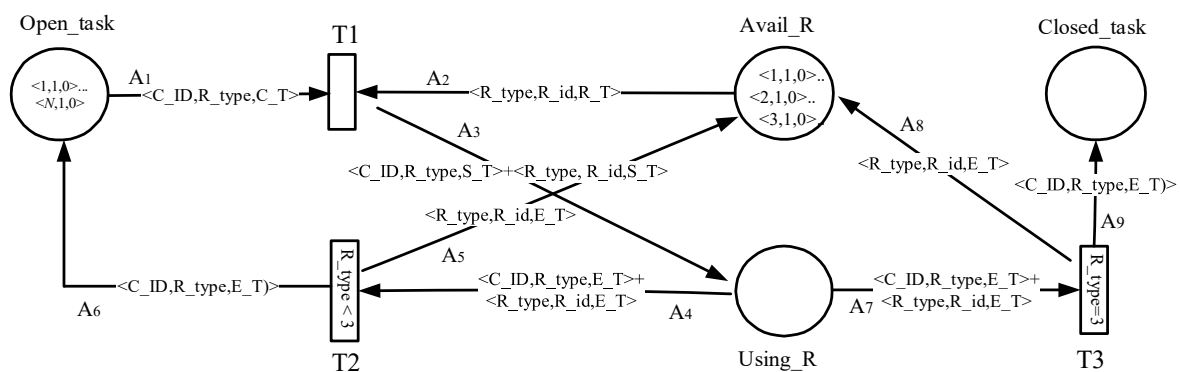


Figure 10. A Predicate/Transition net model for simulation.

$M_0 = \{<1,1,0>, \dots, <N,1,0>\} \cup \{<1,1,0>, \dots, <1,p,0>\} \cup \{<2,1,0>, \dots, <2,k,0>\} \cup \{<3,1,0>, \dots, <3,q,0>\}$ is an initial marking, with the container tokens staying with the Open_task predicate while resource tokens staying with the Avail_R predicate. Each container token contains the three attributes: C_ID (container ID), R_type (resource type), and C_T (available time). Each resource token contains the three attributes: R_type (resource type), R_id (resource id), and R_T (available time). The attribute R_type can have the values 1, 2, and 3, corresponding to a YC, YT, and QC instance, respectively. The summation of labels, $<C_ID, R_type, C_T> + <R_type, R_id, R_T>$ on A3, is a formal sum. To trigger the T1, it requires one container token and one resource token simultaneously. The “R_type < 3” in T2 and “R_type = 3” in T3 are logical formulas (conditions) for triggering the transitions T2 and T3, respectively.

The simulation model works in this way. First, the initial marking M_0 enables the transition T1. The trigger of transition T1 moves one container token $<C_ID, R_type, C_T>$ and one resource token $<R_type, R_id, S_T>$ to the Using_R predicate. The start resource usage time, S_T, is determined by the formula, $Max\{C_T, R_T\}$. The assignment of a container to resource instances is determined by the load-balancing heuristic specified in Step 3 of the method framework. The sequence is determined by the sequencing method specified in Step 4 of the method framework. After a consuming duration at the (timed) predicate Using_R, two scenarios are available. In case that the formula (R_type < 3) in T2 is satisfied, it will trigger the transition T2, which will further transit the container token $<C_ID, R_type, E_T>$ back to the Avail_task predicate, with the value of R_type being added 1 (meaning the change to the next operation); meanwhile the resource token $<R_type, R_id, E_T>$ is returned to the Avail_R predicate. The E_T indicates the end usage time of the resource. In case that the logical formula (R_type = 3) in T3 is satisfied, then it

will trigger the transition T3, which will move the container token $\langle C_ID, R_type, E_T \rangle$ to the Closed_task predicate, meanwhile returning the resource token $\langle R_type, R_id, E_T \rangle$ to the Avail_R predicate. The Closed_task predicate is used to keep all completed container tokens. The triggers of transitions of T1, T2, and T3 will continue until all containers are staying with the Closed_task predicate. This indicates the completion of all containers, and the simulation model then stops running. During the simulation process, the S_T (including S1, S2, and S3) and E_T of each resource usage is recorded and evaluated. The best solution can be identified and output.

5. Numerical Experiments

Based on the method framework proposed in Section 4.1, different methods, including Method1 (SBB), Method2 (GA), Method3 (PSO), and Method4 (MGPSO), have been developed by Java programming language. Experiments were conducted in a personal notebook with an Intel PENTIUM CPU 2117U (64 bits and 1.8 GHz) and 4GB DRAM to investigate the effectiveness of these methods.

5.1. Parameter Values

To test the effectiveness of alternative techniques employed in stage one, further experiments are carried out in this section. Table 2 shows the parameters for YC, YT, and QC.

Table 2. The parameter setting for YC, YT and QC.

YSP		YC		YT		QC		VSP	
X	40	$\ P\ $	2	$\ K\ $	5	$\ Q\ $	2	$\ B\ $	5
Y	10	v_x	1 M/s	$PT2_i$	600 s	v_b	1 M/s	R	5
Z	10	v_y	1 M/s			v_r	1 M/s	T	5
		v_z	1 M/s			v_t	1 M/s		
		H	11			\hat{H}	40 M		

5.2. Experiment Data Generation

Based on the parameter values defined in Table 3, the computer generates the data of container, YSP, and VSP for experiments.

Table 3. The parameter setting for different approaches.

Parameter Values	Method1 (SBB)	Method2 (GA)	Method3 (PSO)	Method4 (MGPSO)
$\ T\ $	10, 20, 40, 80	10, 20, 40, 80	10, 20, 40, 80	10, 20, 40, 80
P	120	120	120	120
GN(t)	-	-	-	Equation (33)
n	-	-	-	P/GN(t)
n_ls	-	-	-	2
T	1	500	500	250
w	-	-	0.5	-
\bar{V}	-	-	2	-
\underline{V}	-	-	-2	-
Rm	-	0.3	-	0.3
Rc	-	0.4	-	-

$\|T\|$: number of tasks (containers); P: total number of populations; GN(t): number of groups; n: number of particles in each group; n_ls: number of local searches; T: total number of iterations; \bar{V} : low limit of velocity; Rm: mutation rate; Rc: crossover rate; and -: not used.

5.3. An Example of a Small-Sized Problem

A small-sized problem of 10×10 ($N = 10$) is used to illustrate the solution obtained from the Method4 (MGPSO).

5.3.1. Input Data

Table 4 shows the YSP data of export containers in a storage block. Row 1 indicates the container No; Row 2 shows the type of container (type = 2 means an export container); the Rows 3 to 5 show the coordinates of x , y , and z in the block, which correspond to the bay, row, and tier numbers of a container, respectively, of this container.

Table 4. The YSP data of export containers.

j	1	2	3	4	5	6	7	8	9	10
type	2	2	2	2	2	2	2	2	2	2
x	23	3	31	10	38	30	20	5	28	33
y	7	7	9	1	10	4	1	8	2	9
z	8	10	8	5	4	7	5	2	2	2

Table 5 shows the VSP data of export containers in a vessel. Row 1 indicates container No.; Rows 2, 3, and 4 show the bay, row, and tier numbers, respectively, for this container. Table 6 shows the cost matrix C_{ji} automatically generated by the computer based on the VSP data (Table 5). From which, we find the VSP sequence constraints.

1. Container 1 should be loaded before Container 5.
2. Container 2 should be loaded before Container 4.
3. Container 4 should be loaded before Container 7.

Table 5. The VSP data of export containers.

j	1	2	3	4	5	6	7	8	9	10
type	2	2	2	2	2	2	2	2	2	2
b	3	2	4	2	3	3	2	2	4	3
r	3	4	4	4	3	2	4	3	5	4
t	1	2	2	3	3	4	5	2	1	2

Table 6. The cost matrix (C_{ij}) of VSP constraints.

$i \backslash j$	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	600	0	0	0	0	0	0	0	0
5	600	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	600	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

A penalty cost of $PC = 600$ s is imposed for each violation of the above constraints.

5.3.2. Output Result

Table 7 shows the best solution found by Method4 (MGPSO). Row 2 indicates the YC assigned for the container i ; Row 3 shows the order of container i on its assigned YC; the Rows 4 and 5, respectively, show the start and end using times on the assigned YCs; Row 6 shows the YT assigned to the container i ; the Rows 7 and 8, respectively, show the start and end using time of the YT assigned; Row 9 shows the QC assigned to the container i ; Row 10 shows the loading order of the QC assigned; the Rows 11 and 12, respectively, show

the start and end using times the assigned QC. Table 7 shows that the best Z found by the MGPSO is 1627.5 s. The best solution is found in Iteration 2.

Table 7. The solutions to the integrated problem of YCSP, YTSP, and QCSP found by Method4 (MGPSO).

j	1	2	3	4	5	6	7	8	9	10
YC	1	1	2	1	2	2	1	1	2	2
Seq.	4	1	5	3	4	1	5	2	3	2
S1	229.2	0	465.6	156	348	0	331.2	62.4	258	157.2
E1	331.2	62.4	550.8	229.2	465.6	157.2	400.8	156	348	258
YT	1	1	5	4	4	3	3	2	2	5
S2	662.4	62.4	858	229.2	829.2	157.2	757.2	156	756	258
E2	1262.4	662.4	1458	829.2	1429.2	757.2	1357.2	756	1356	858
QC	1	1	2	1	1	1	1	1	2	1
Seq.	6	1	2	4	8	3	7	2	1	5
S3	1263.2	662.4	1488.8	1011.1	1505.6	894	1394.6	773.4	1356	1131.7
E3	1394.6	773.4	1626.4	1131.7	1627.5	1011.1	1505.6	894	1488.8	1263.2

The results are found to be a workload balanced in terms of the number of containers assigned, with constraints being also considered. For YCs, Containers 1,2,4,7, and 8 are assigned to the YC 1 with the loading sequence 2-8-4-1-7, while Containers 3,5,6,9, and 10 are assigned to YC 2 with the loading sequence 6-10-9-5-3. The $S1$ and $E1$ are start and end using times for each assigned YC. For the YTs, Containers 2,8,4,1, and 7 are assigned to YTs 1,2,4,1, and 3, respectively; while Containers 6,10,9,5, and 3 are assigned to YTs 3,5,2,4, and 5, respectively. These YTs are assigned and running in a loop; each YT serves two containers. The $S2$ and $E2$, respectively, indicate the start and end using times for each assigned YT. For QCs, Containers 1,2,4,5,6,7,8, and 10 are assigned to QC 1 with the loading sequence 2-8-6-4-10-1-7-5 into the ship, while Containers 3 and 9 are assigned to QC 2 with the loading sequence 9-3. The two loading sequences of containers are found conforming to the VSP constraints. The $S3$ and $E3$ indicate the start and end using times of each assigned QC, respectively. The assignments for QC 1 and QC 2 appear to be unbalanced. This is because of the need to conform to the constraint, i.e., containers of the same bay are assigned to the same QC. Specifically, it starts assigning Containers 2,8,4,7 (in Bay 2) and Container 10 (in Bay 3) to QC 1. However, it is found that Containers 1, 5, and 6 are all in Bay 3; thus, these containers have to be assigned to QC 1 continuously. Following this, the remaining containers, including Containers 9 and 3 (in Bay 4), are assigned to QC 2. It is noted that the end using times of QC 1 and QC 2 are 1627.5 and 1626.4, respectively, which is a desirable result. As QC 1 works in the bay space consisting of Bays 2 and 3, and QC 2 works in the bay space consisting of Bay 4, the two QCs are not interfering each other.

5.4. Extensive Experiments

Extensive experiments have been conducted, and the results are shown in Table 8. The results are described below.

- The experimental results of the problem size 10×10 show that Method4 (MGPSO) outperforms Method3 (PSO), Method2 (GA), and Method1 (SBB) with the edges 26.3%, 25.4%, and 28.0%, respectively, in terms of makespan.
- The experimental results of the problem size 20×20 show that Method4 (MGPSO) outperforms Method3 (PSO), Method2 (GA), and Method1 (SBB) with the edges 25.3%, 26.8%, 28.9%, respectively, in terms of makespan.
- The experimental results of the problem size 40×40 show that the Method4 (MGPSO) outperforms Method3 (PSO), Method2 (GA), and Method1 (SBB) with the edges 25.8%, 26.2%, and 28.4%, respectively, in terms of makespan.
- The experimental results of the problem size 80×80 show that Method4 (MGPSO) outperforms Method3 (PSO), Method2 (GA), and Method1 (SBB) with the edges 26.9%, 27.3%, and 31.8%, respectively, in terms of makespan.

Table 8. The results obtained from different approaches under different problem sizes.

No.	Method1 (SBB)				Method2 (GA)				Method3 (PSO)				Method4 (MGPSO)		
	Z (s)	T (s)	IT	G (%)	Z(s)	T(s)	IT	G (%)	Z (s)	T (s)	IT	G (%)	Z (s)	T (s)	IT
10 × 10															
1	2050.5	0.05	1	26.4	2152.9	1134	100	32.7	2079.3	78	7	28.1	1622.8	366	1
2	2079.6	0.05	1	28.6	1921.6	133	1	18.8	1964.8	76	1	21.5	1616.9	365	1
3	2129.8	0.05	1	21.6	2132.2	130	162	21.7	2175.4	77	2	24.2	1751.6	360	1
4	2138.3	0.05	1	30.0	2118.2	131	342	28.7	2108.6	78	28	28.2	1645.4	389	1
5	2058.2	0.05	1	27.5	1957.3	134	5	21.2	2000.5	77	9	23.9	1614.4	354	1
6	2128.8	0.05	1	31.6	2051.4	131	5	26.9	2071.3	76	6	28.1	1617.1	349	1
7	2058.4	0.05	1	26.4	2093.5	131	1	28.5	2036.7	76	1	25.0	1628.9	365	6
8	2011.1	0.05	1	25.3	2083.3	134	3	29.8	2166.6	76	223	35.0	1604.8	362	4
9	2160.3	0.05	1	32.8	2060.3	131	2	26.7	2080.0	76	4	27.9	1626.7	349	2
10	2234.5	0.05	1	30.6	2041.1	132	17	19.3	2084.3	78	8	21.9	1710.3	374	1
Avg.	2104.9	0.05	1	28.0	2061.2	232.1	64	25.4	2076.8	76.8	29	26.3	1643.9	363.3	2
20 × 20															
1	3571.6	0.08	1	28.1	3582.0	257	278	28.5	3356.0	149	35	20.4	2787.9	662	230
2	3724.5	0.08	1	33.5	3626.4	258	213	30.0	3568.7	147	56	27.9	2790.5	690	146
3	3090.9	0.08	1	9.6	3447.2	252	164	22.3	3465.5	150	97	22.9	2819.2	696	7
4	3884.5	0.08	1	38.2	3373.0	257	450	20.0	3394.4	152	383	20.8	2810.9	717	31
5	3723.8	0.08	1	33.7	3595.9	259	410	29.1	3603.6	148	9	29.4	2785.5	707	70
6	3208.5	0.08	1	14.4	3526.6	261	9	25.8	3693.3	102	261	31.7	2803.5	683	55
7	3595.3	0.08	1	29.1	3627.0	262	317	30.2	3376.7	148	60	21.2	2785.5	718	147
8	3799.5	0.08	1	36.1	3350.7	262	309	20.0	3368.2	148	22	20.6	2792.7	698	5
9	3585.1	0.08	1	27.2	3588.8	259	384	27.4	3636.3	147	19	29.0	2818.0	691	8
10	3890.1	0.08	1	39.1	3768.3	259	213	34.7	3600.7	146	18	28.7	2797.5	681	174
Avg.	3607.4	0.08	1	28.9	3548.6	258.6	275	26.8	3506.3	143.7	96	25.3	2799.1	694.3	87
40 × 40															
1	6697.5	0.08	1	28.6	6909.1	509	271	32.6	6895.6	285	31	32.4	5209.6	1167	110
2	6751.7	0.08	1	30.6	6627.7	491	204	28.2	5936.2	279	173	14.8	5169.9	1297	47
3	6694.7	0.08	1	29.4	6654.6	495	45	28.6	6489.6	286	330	25.4	5174.7	1258	92
4	6580.2	0.08	1	26.9	6217.7	499	94	20.0	6344.9	285	340	22.4	5183.3	1211	91
5	6689.7	0.08	1	29.4	6389.4	501	35	23.6	6327.9	281	422	22.4	5167.9	1373	42
6	6586.0	0.08	1	26.9	6671.3	503	153	28.5	6688.9	285	497	28.9	5190.2	1227	219
7	6592.3	0.08	1	26.8	6485.1	503	67	24.7	6621.2	285	121	27.4	5198.7	1312	4
8	6796.8	0.08	1	30.9	6719.2	501	238	29.4	6518.8	282	50	25.5	5194.0	1309	28
9	6570.3	0.08	1	27.0	6551.1	497	419	26.7	6743.2	288	357	30.4	5172.2	1364	116
10	6595.4	0.08	1	27.4	6217.4	495	458	20.1	6665.3	287	438	28.7	5177.0	1316	117
Avg.	6655.5	0.1	1	28.4	6544.3	499.4	198	26.2	6523.2	284.3	276	25.8	5183.8	1283.4	87
80 × 80															
1	13,038.4	0.09	1	30.6	12,483.9	992	272	25.1	12,444.1	566	272	24.7	9982.2	2576	56
2	12,278.7	0.09	1	23.0	12,733.5	908	103	27.5	12,493.5	555	199	25.1	9986.7	2471	131
3	13,054.0	0.09	1	30.6	12,850.4	989	37	28.6	12,478.4	551	226	24.8	9996.4	2398	116
4	13,619.6	0.09	1	36.4	12,699.8	960	1	27.2	12,886.8	556	348	29.1	9985.7	2421	161
5	13,257.0	0.09	1	33.0	12,514.7	979	203	25.6	12,604.7	548	314	26.5	9965.2	2315	73
6	13,507.1	0.09	1	35.3	12,895.5	945	266	29.2	12,717.8	626	85	27.4	9983.2	2423	15
7	12,538.6	0.09	1	25.6	12,593.6	984	178	26.2	12,182.4	566	252	22.0	9981.8	2484	95
8	14,272.8	0.09	1	43.0	12,776.7	983	114	28.0	12,958.4	566	126	29.8	9983.1	2463	74
9	13,023.6	0.09	1	30.5	12,560.4	977	433	25.8	12,903.2	562	140	29.3	9980.8	2565	157
10	12,932.9	0.09	1	29.8	12,947.2	964	107	29.9	12,945.4	558	212	29.9	9963.8	2283	105
Avg.	13,152.3	0.1	1	31.8	12,705.6	968.1	171	27.3	12,661.5	565.4	217	26.9	9980.9	2439.9	98

Z: the objective function value; T: time (s); IT: the iteration of the best solution; and G: gap.

5.5. Statistically Test

The *t*-test, referring to [45], is used to test the results at the significance level $\alpha = 0.05$. Hypotheses H0 and H1 are set. Ho is a null hypothesis that assumes that the average Z values obtained from different methods are not different; H1 is the hypothesis that assumes that the average Z values obtained from different methods are different.

Table 9 shows the statistical test results obtained from different methods comparing to Method4 (MGPSO). The symbol “+” indicates that Method4 (MGPSO) is better; the symbol “−” indicates that Method4 (MGPSO) is inferior; the symbol “N” indicates that there is no difference between Method4 (MGPSO) and the comparing method. Since the *p*-values of all pairs of comparison are ≤ 0.005 , it leads to the conclusion that Method4 (MGPSO) outperforms the other methods. For example, the *p*-values of the problem size 10×10 when comparing Method4 (MGPSO) with Method1 (SBB), Method2 (GA), and Method3

(PSO) are found being 5.20695E-10, 2.51553E-08, and 2.90149E-09, respectively. This leads to the rejection of H0 and the acceptance of H1.

Table 9. Average Z-values and their *t*-test results obtained from the four approaches under different problem sizes.

Problem Size	Method1 (SBB)		Method2 (GA)		Method3 (PSO)		Method4 (MGPSO)
	Avg. Z	<i>t</i> -test	Avg. Z	<i>t</i> -test	Avg. Z	<i>t</i> -test	Avg. Z
10 × 10	2104.9	+	2061.2	+	2076.8	+	1643.9
20 × 20	3607.4	+	3548.6	+	3506.3	+	2799.1
20 × 40	6655.5	+	6544.3	+	6523.2	+	5183.8
80 × 80	13,152.3	+	12,705.6	+	12,661.5	+	9980.9

6. Results Analysis and Discussions

The results are discussed accordingly, as follows.

- The research aims to improve operational efficiency for a container terminal by coordinating YC, YT, and QC. The VSP constraints are especially respected, and they can affect the QC operation. The penalty costs of violating the VSP constraint is considered in the objective function with the purpose of exclude undesired solutions. The small-sized experiment has demonstrated the found solution, which is found to be free from violating all the VSP constraints.
- As QC is bigger and heavier, it requires more setup time and tends to become a bottleneck in the container terminal operation. Therefore, coordinating YC and YT operations to facilitate the QC operation is reasonable.
- The load-balancing concept enables better utilization of available resources. This concept has been applied to all equipment, including YC, YT, and QC, in this research. The small-sized experiment shows that the completion of QC 1 and QC 2 in the found solution is close, which implies a good result.
- This research has taken two storage positions of containers, one in the storage block and another in the vessel, into consideration. The position in the block is specified by the YSP, while the position in the vessel is specified by the VSP. Changing positions between the block and vessel require the cooperation of YC, YT, and QC; thus, coordination on these MHE is important. In this research, simulation-based optimization methods are used as the planning tool.
- Method1 (SBB) is simply due to the use of one single iteration. It generates a simple solution. This method makes a YC work in one direction, from low bay to high bay number. Though it facilitates YC operation, it cannot make sure of the benefit to the QC operation. The experimental results show that Method1 is inferior to other methods in terms of makespan.
- Method4 (MGPSO) is found able to find a good solution at an earlier iteration. This advantage is considered owing to its novel features, such as multiple groups, particle reshuffling, adaptive velocity, and decreasing group number. Not only to be more diversified, but particles can also use smart movements, such as *Tabu fly* and *Neighborhood search*, to better search a solution space.
- Unlike the PSO, the MGPSO employs multiple groups of particles so that particles can be more diversified due to being influenced by more elites (i.e., the best particles of groups). This helps avoid particles from being trapped in local optima.
- Figure 11 shows the Z value trends of the problem size 20 × 20 obtained from different methods. Method4 (MGPSO) is found to outperform Method3 (PSO), Method2 (GA), and Method1 (SBB). Method1 (SBB) finds one Z value due to its simplicity. In contrast, Method2 (GA), Method3 (PSO), and Method4 (MGPSO) have evolutionary capabilities. In addition, the evolutionary methods dive to the bottom quickly, and since then, solution improvement is found to not be significant. One likely reason is the use of the

load-balancing concept, which gives a good initial foundation for developing solutions so that the contributions from the sequencing methods become non-significant. This finding suggests that a small iteration run can be employed to save the computational times required if necessary.

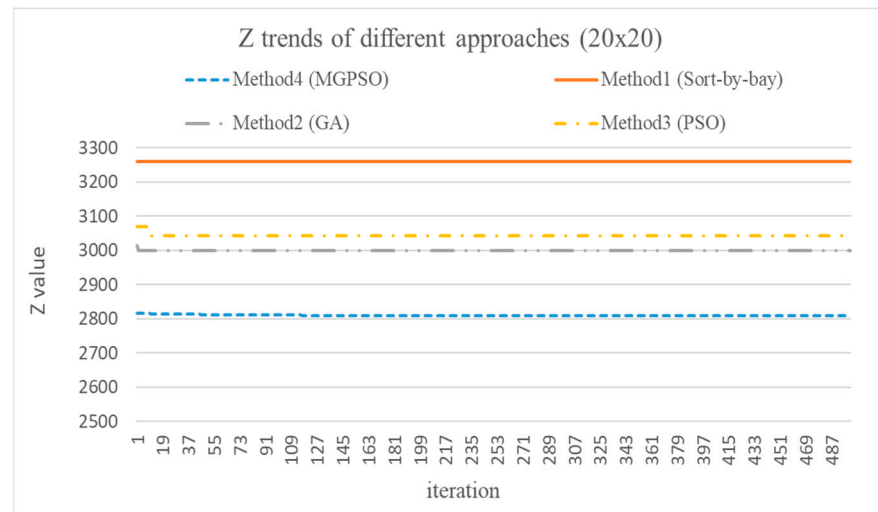


Figure 11. Z trends of different approaches (the problem size 20×20).

7. Conclusions

This research is devoted to solving the YCSP, YTSP, and QCSP simultaneously, with YSP and VSP data taken into account. A MILP has been formulated with the objective aimed at minimizing the makespan. However, due to NP-hard, a framework for developing simulation-based optimization methods has been proposed. Based on this framework, four different methodologies, Method1 (SBB), Method2 (GA), Method3 (PSO), and Method4 (MGPSO), have been developed. A small-sized experiment has been used to illustrate the solution found by Method4 (MGPSO). In addition, more experiments of different problem sizes, including 10×10 , 20×20 , 40×40 , and 80×80 , have been conducted. The experimental results showed that Method4 (MGPSO) outperforms the others. Statically, *t*-tests have been performed to test the robustness of the experiments. This research is considered with the following contributions: (1) the formulation of the MILP model; (2) the framework proposed for the development of simulation-based optimization methods; (3) the development of an improved version of standard PSO, i.e., the MGPSO; and (4) the performance of experiments to investigate these developed methods.

The following research directions can be considered: (1) in addition to the export containers, import containers can be considered simultaneously; (2) this study can be extended to include multiple storage blocks or even to multiple ships; and (3) the uses of other methods, such as firefly algorithm (FA), Bee-inspired Algorithms (BA), Bacterial Foraging Optimization (BFO), Firefly Algorithms (FA), Fish Swarm Optimization (FWO) and Cuckoo Search Algorithm (CSA) as the sequencing methods, can be investigated in the future research.

Author Contributions: Conceptualization, H.-P.H.; Data curation, H.-P.H.; Formal analysis, H.-P.F.; Funding acquisition, H.-P.H.; Investigation, C.-N.W.; Methodology, H.-P.H.; Project administration, C.-N.W.; Software, H.-P.H.; Validation, H.-P.H. and T.-T.D.; Writing-original draft, H.-P.H. and H.-P.F.; Writing-review and editing, C.-N.W., H.-P.F., and T.-T.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and Technology of Taiwan under the grant MOST 110-2410-H-992-037, and partially supported by the National Kaohsiung University of Science and Technology under the project number 110G02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors appreciate the support from the National Kaohsiung University of Science and Technology, Ministry of Science and Technology in Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. UNCTAD. United Nations Conference on Trade and Development-Review of Maritime Transport. 2019. Available online: <http://unctad.org/en/publicationsLibrary/rmt2019en.pdf> (accessed on 15 June 2021).
2. Stahlbock, R.; Steenken, D. Container terminal operation and operations research—A classification and literature review. *OR Spectr.* **2004**, *26*, 3–49. [[CrossRef](#)]
3. Bierwirth, C.; Meisel, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2010**, *202*, 615–627. [[CrossRef](#)]
4. Hsu, H.-P.; Wang, C.-N. Resources Planning for Container Terminal in a Maritime Supply Chain Using Multiple Particle Swarms Optimization (MPSO). *Mathematics* **2020**, *8*, 764. [[CrossRef](#)]
5. Imai, A.; Chen, H.C.; Nishimura, E.; Papadimitriou, S. The simultaneous berth and quay crane allocation problem. *Transp. Res. Part E Logist. Transp. Rev.* **2008**, *44*, 900–920. [[CrossRef](#)]
6. Hsu, H.-P. A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously. *Swarm Evol. Comput.* **2016**, *27*, 156–168. [[CrossRef](#)]
7. Raa, B.; Dullaert, W.; Van Schaeren, R. An enriched model for the integrated berth allocation and quay crane assignment problem. *Expert Syst. Appl.* **2011**, *38*, 14136–14147. [[CrossRef](#)]
8. Carlo, H.J.; Vis, I.F.; Roodbergen, K.J. Storage yard operations in container terminals: Literature overview, trends, and research directions. *Eur. J. Oper. Res.* **2014**, *235*, 412–430. [[CrossRef](#)]
9. Kizilay, D.; Eliyi, D.T. A comprehensive review of quay crane scheduling, yard operations and integrations thereof in container terminals. *Flex. Serv. Manuf. J.* **2020**, *33*, 1–42. [[CrossRef](#)]
10. Azevedo, A.T.; Salles-Neto, L.; Chaves, A.; Moretti, A.C. Solving the 3D stowage planning problem integrated with the quay crane scheduling problem by representation by rules and genetic algorithm. *Appl. Soft Comput.* **2018**, *65*, 495–516. [[CrossRef](#)]
11. Zheng, K.; Lu, Z.; Sun, X. An Effective Heuristic for the Integrated Scheduling Problem of Automated Container Handling System Using Twin 40' Cranes. In Proceedings of the Second International Conference on Computer Modeling And Simulation, Sanya, China, 22–24 January 2010; Volume 1, pp. 406–410. [[CrossRef](#)]
12. Fazli, M.; Fathollahi-Fard, A.M.; Tian, G. Addressing a Coordinated Quay Crane Scheduling and Assignment Problem by Red Deer Algorithm. *Int. J. Eng.* **2019**, *32*, 1186–1191. [[CrossRef](#)]
13. Sun, D.; Tang, L.; Baldacci, R. A Benders decomposition-based framework for solving quay crane scheduling problems. *Eur. J. Oper. Res.* **2018**, *273*, 504–515. [[CrossRef](#)]
14. Huang, S.Y.; Li, Y. A bounded two-level dynamic programming algorithm for quay crane scheduling in container terminals. *Comput. Ind. Eng.* **2018**, *123*, 303–313. [[CrossRef](#)]
15. Rouky, N.; Abourraja, M.N.; Boukachour, J.; Boudebous, D.; Alaoui, A.E.H.; El Khoukhi, F. Simulation optimization based ant colony algorithm for the uncertain quay crane scheduling problem. *Int. J. Ind. Eng. Comput.* **2019**, 111–132. [[CrossRef](#)]
16. Nielsen, I.E.; Do, N.A.D.; Nguyen, V.D.; Nielsen, P.; Michna, Z. Reducing Truck Emissions in Import Operations at Container Terminal—A Case Study in a Singaporean Port. In *Technology Management for Sustainable Production and Logistics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 133–151. [[CrossRef](#)]
17. Kim, K.H.; Bae, J.W. A Look-Ahead Dispatching Method for Automated Guided Vehicles in Automated Port Container Terminals. *Transp. Sci.* **2004**, *38*, 224–234. [[CrossRef](#)]
18. Bish, E.K.; Chen, F.Y.; Leong, Y.T.; Nelson, B.L.; Ng, J.W.C.; Simchi-Levi, D. Dispatching vehicles in a mega container terminal. In *Container Terminals and Cargo Systems*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 179–194. [[CrossRef](#)]
19. Ng, W.C.; Mak, K.L.; Zhang, Y.X. Scheduling trucks in container terminals using a genetic algorithm. *Eng. Optim.* **2007**, *39*, 33–47. [[CrossRef](#)]
20. Ng, W.; Mak, K. Yard crane scheduling in port container terminals. *Appl. Math. Model.* **2005**, *29*, 263–276. [[CrossRef](#)]
21. Jung, S.H.; Kim, K.H. Load scheduling for multiple quay cranes in port container terminals. *J. Intell. Manuf.* **2006**, *17*, 479–492. [[CrossRef](#)]
22. He, J.; Chang, D.; Mi, W.; Yan, W. A Strategy for Yard Crane Scheduling Based on Hybrid Parallel Genetic Algorithm. In Proceedings of the 2008 International Symposium on Knowledge Acquisition and Modeling, Wuhan, China, 21–22 December 2008; pp. 678–683. [[CrossRef](#)]
23. Javanshir, H.; SEYED, A.G.S. Yard crane scheduling in port container terminals using genetic algorithm. *Int. J. Ind. Eng.* **2010**, *6*, 39–50.

24. Chang, D.; Jiang, Z.; Yan, W.; He, J. Developing a dynamic rolling-horizon decision strategy for yard crane scheduling. *Adv. Eng. Inform.* **2011**, *25*, 485–494. [[CrossRef](#)]
25. He, J.; Huang, Y.; Yan, W. Yard crane scheduling in a container terminal for the trade-off between efficiency and energy consumption. *Adv. Eng. Inform.* **2015**, *29*, 59–75. [[CrossRef](#)]
26. Tsai, F.-M.; Lu, C.-C.; Chang, Y.-M. A network model for solving the yard truck routing and scheduling problem. *Int. J. Logist. Manag.* **2016**, *27*, 353–370. [[CrossRef](#)]
27. Ng, W. Crane scheduling in container yards with inter-crane interference. *Eur. J. Oper. Res.* **2005**, *164*, 64–78. [[CrossRef](#)]
28. Sharif, O.; Huynh, N. Yard crane scheduling at container terminals: A comparative study of centralized and decentralized approaches. *Marit. Econ. Logist.* **2012**, *14*, 139–161. [[CrossRef](#)]
29. Guo, X.; Huang, S.Y.; Hsu, W.J.; Low, M.Y.H. Dynamic yard crane dispatching in container terminals with predicted vehicle arrival information. *Adv. Eng. Inform.* **2011**, *25*, 472–484. [[CrossRef](#)]
30. Li, W.; Wu, Y.; Goh, M. A Continuous-Time Model for Multiple Yard Crane Scheduling with Last-Minute Job Arrivals. In *Planning and Scheduling for Maritime Container Yards*; Springer: Cham, Switzerland, 2015; pp. 53–83. [[CrossRef](#)]
31. Chen, L.; Bostel, N.; Dejax, P.; Cai, J.; Xi, L. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *Eur. J. Oper. Res.* **2007**, *181*, 40–58. [[CrossRef](#)]
32. Cao, J.X.; Lee, D.H.; Chen, J.H.; Shi, Q. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 344–353. [[CrossRef](#)]
33. Chen, L.; Langevin, A.; Lu, Z. Integrated scheduling of crane handling and truck transportation in a maritime container terminal. *Eur. J. Oper. Res.* **2013**, *225*, 142–152. [[CrossRef](#)]
34. Wu, Y.; Luo, J.; Zhang, D.; Dong, M. An integrated programming model for storage management and vehicle scheduling at container terminals. *Res. Transp. Econ.* **2013**, *42*, 13–27. [[CrossRef](#)]
35. Xue, Z.; Zhang, C.; Miao, L.; Lin, W.-H. An ant colony algorithm for yard truck scheduling and yard location assignment problems with precedence constraints. *J. Syst. Sci. Syst. Eng.* **2013**, *22*, 21–37. [[CrossRef](#)]
36. He, J.; Huang, Y.; Yan, W.; Wang, S. Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption. *Expert Syst. Appl.* **2015**, *42*, 2464–2487. [[CrossRef](#)]
37. Luo, J.; Wu, Y.; Mendes, A.B. Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container terminal. *Comput. Ind. Eng.* **2016**, *94*, 32–44. [[CrossRef](#)]
38. Kizilay, D.; Eliiyi, D.T.; Van Hentenryck, P. Constraint and Mathematical Programming Models for Integrated Port Container Terminal Operations. In *Proceedings of the International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Delft, The Netherlands, 26–29 June 2018; Springer: Cham, Switzerland, 2018; pp. 344–360. [[CrossRef](#)]
39. Yang, Y.; Zhong, M.; Dessouky, Y.; Postolache, O. An integrated scheduling method for AGV routing in automated container terminals. *Comput. Ind. Eng.* **2018**, *126*, 482–493. [[CrossRef](#)]
40. Jonker, T.; Duinkerken, M.B.; Yorke-Smith, N.; de Waal, A.; Negenborn, R.R. Coordinated optimization of equipment operations in a container terminal. *Flex. Serv. Manuf. J.* **2019**, *33*, 281–311. [[CrossRef](#)]
41. Yang, Y.; Zhu, X.; Haghani, A. Multiple Equipment Integrated Scheduling and Storage Space Allocation in Rail–Water Intermodal Container Terminals Considering Energy Efficiency. *Transp. Res. Rec. J. Transp. Res. Board* **2019**, *2673*, 199–209. [[CrossRef](#)]
42. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95—International Conference on Neural Networks*, Perth, Australia, 27 November–1 December 1995. [[CrossRef](#)]
43. Hsu, H.P. A fuzzy knowledge-based disassembly process planning system based on fuzzy attributed and timed predicate/transition net. In *Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Budapest, Hungary, 9–12 October 2016; Volume 47, pp. 1800–1813.
44. Hsu, H.-P.; Wang, C.-N.; Chou, C.-C.; Lee, Y.; Wen, Y.-F. Modeling and Solving the Three Seaside Operational Problems Using an Object-Oriented and Timed Predicate/Transition Net. *Appl. Sci.* **2017**, *7*, 218. [[CrossRef](#)]
45. Zhang, Y.; Gong, D.-W.; Cheng, J. Multi-Objective Particle Swarm Optimization Approach for Cost-Based Feature Selection in Classification. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2015**, *14*, 64–75. [[CrossRef](#)]