Container batch grouping for automated container terminals

Lijun Yue
Shijiazhuang Tiedao University, Shijiazhuang, China, and
Jasmine Siu Lee Lam
Technical University of Denmark, Lyngby, Denmark

Received 26 January 2025
Revised 14 April 2025
Accepted 26 May 2025

Abstract

Purpose – In automated container terminals, mutual waiting between yard cranes (YCs) and automated guided vehicles (AGVs) causes resource wastage, extended task execution times and traffic congestion. This study proposes a strategy called batch grouping, which classifies containers by size, weight and destination for interchangeable handling, effectively reducing waiting times.

Design/methodology/approach – A mixed-integer programming model is developed to minimize YC costs by incorporating constraints on the number of similar containers to be handled at each time point into the classic YC scheduling optimization model. Additionally, a mixed-integer programming model is constructed to minimize AGV costs, which includes AGV capacity constraints based on the classic AGV scheduling optimization model. A hybrid heuristic algorithm and GUROBI software are used to solve these models.

Findings – Experimental results show that integrating classification and interchangeability into the joint scheduling of YCs and AGVs achieves a cost reduction of up to 8.9%.

Practical implications – Terminal operators can implement these findings to streamline their scheduling processes, leading to improved efficiency and reduced traffic congestion within the terminal environment.

Originality/value – This study contributes to the field of terminal operations by introducing an innovative batch grouping strategy that addresses the critical issue of mutual waiting between YCs and AGVs.

Keywords Port, Automated container terminal, Automated guided vehicle, Non-cantilever yard crane, Container classification, Heuristic algorithm

Paper type Research paper

1. Introduction

Containers are extensively utilized in the international trade and logistics sector. The operations of automated container terminals (ACTs) facilitate a continuous and efficient flow of container transportation and handling (Lam and Chen, 2021). The large-scale development of container ships has accelerated, which increases the demand for container terminal operations. To meet these demands, many ports have expedited the construction of automated terminals and optimized the allocation and scheduling of operational resources to improve efficiency, ensuring that ships can handle imports and exports within the specified time and depart as scheduled. Increased ship operation time in port can result from congestion, declining operational efficiency or cascading events, all of which directly impact port efficiency (Dragović *et al.*, 2023). This presents a significant challenge in optimizing resource use and equipment interaction. To ensure timely departures, it is crucial to schedule quay cranes (QCs), yard cranes (YCs) and automated guided vehicles (AGVs) effectively, enhancing operational efficiency and reducing costs. Unlike manual container terminals, automated ones cannot coordinate operations through truck drivers or crane operators, so delays in container handover can prolong berthing times. To mitigate this, classifying containers by attributes such as destination, size and weight and exchanging similar types can reduce container re-handling and minimize waiting times for AGVs and YCs. Thus, optimizing the scheduling of YCs and AGVs in automated terminals, considering multiple container attributes, is essential for improving operational efficiency and is a key concern for terminal managers.

This work is partly supported by the China Scholarship Council Fund (CSC202206570020) and Natural Science Foundation of Hebei Province, China (G2025210006).

Maritime Business Review
Vol. 10 No. 3, 2025
pp. 210-225
Emerald Publishing Limited
e-ISSN: 2397-3765
p-ISSN: 2397-3757
DOI 10.1108/MABR-01-2025-0008

Optimizing resource allocation is an effective approach for container terminals to reduce costs and enhance efficiency. Reducing waiting times and balancing the operational loads of YCs and AGVs are key focuses in current research on their joint scheduling problems. To reduce waiting times and balance workload, Xing *et al*. (2023) developed a model adjusting AGV speeds to minimize YC completion time and AGV energy use, solving it with directed graphs and heuristics. Xu *et al*. (2021) synchronized AGV and YC speeds to eliminate wait times. Chen *et al*. (2020) constructed a network flow model to balance the workloads of YCs and AGVs, using an alternating direction multiplier algorithm. Hsu *et al*. (2021) developed separate optimization models for YC and trucks, employing a heuristic for load balancing and simulation for interaction adjustment. Tao *et al*. (2023) introduced a load balancing agreement to reduce traffic congestion by optimizing travel distance and workload imbalance with a mixed-integer model and proposed the NSGA-II algorithm to solve it.

Several studies have independently investigated the optimization of scheduling for QCs (Kong *et al*., 2024), YCs (Oladugba *et al*., 2023) and horizontal transport vehicles (Chen *et al*., 2021). Yue *et al.* (2023) developed a mixed-integer model for dual YC to minimize AGV wait times and maximum completion time using a rule-based heuristic algorithm. Vallada *et al*. (2023) created a yard allocation and single crane scheduling model to minimize weighted costs by solving it with a local search algorithm. For AGV scheduling, Zhong *et al*. (2020) built a model to minimize AGV delay time, solving it with adaptive particle swarm optimization (PSO). Wang and Zeng (2022) designed a branch-and-bound algorithm to solve the sequencing for AGVs and proposed a heuristic algorithm to generate conflict-free routing. Branch-and-bound is often used for solving small-scale foundational problems. In practice, the joint scheduling optimization problem between AGVs and gantry cranes is large and complex, leading researchers to use heuristic algorithms to approximate optimal solutions (Naeem *et al*., 2023).

In summary, we found these literature gaps: (1) Existing research on the joint scheduling of YCs and AGVs treats all containers to be import and export as heterogeneous, without considering that the order of operation for containers of the same type in terms of destination port, size and weight can reduce the waiting time between YCs and AGVs. (2) Research on considering buffer brackets only simplifies them as fixed time window constraints, but the limited capacity of buffer brackets may be occupied for a long time by containers that do not arrive/leave within the time window and changing the order of operation for containers of the same type can promote the mobility of containers on the buffer brackets. (3) Research on AGV or YC scheduling mostly aims to minimize completion time. However, the double-cycle heavy load operation of AGVs or YCs (round-trip heavy load operation) can improve their utilization and reduce costs and changing the order of operation for containers of the same type can increase the number of double-cycle heavy load operations for YCs and AGVs.

Therefore, we propose a strategy called batch grouping, which aims at reducing inter-equipment waiting times by categorizing containers and optimizing the operational sequence of homogeneous containers. Containers of the same type can be interchanged and the time required for AGVs and YCs to load and unload containers varies by location. By optimizing the sequence of loading and unloading these containers, waiting times can be effectively reduced.

2. Problem description

A shipping company classifies containers based on attributes such as shipper, vessel name, voyage number, destination port, dimensions and weight and develops a loading plan accordingly. Upon the vessel's arrival at a port, the terminal allocates berths and schedules for QCs, YCs and AGVs based on this loading plan. Delays in AGV operations may lead to QC delays, potentially extending ship berthing times and causing broader scheduling disruptions. For instance, as shown in Figure 1, Type III containers are designated for Quay Crane 1, while

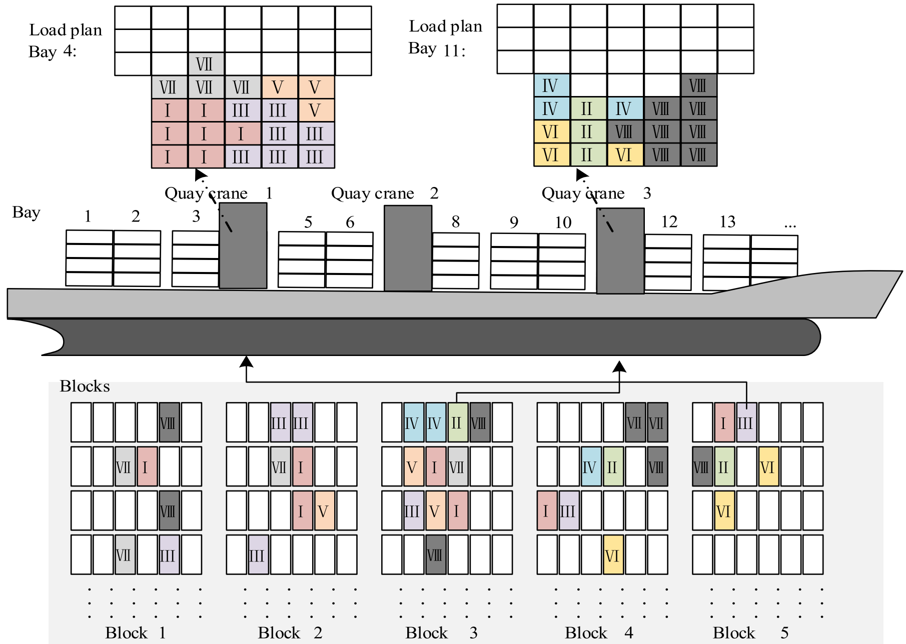

Figure 1. Quay crane handling plan for classified container. **Source(s):** Figure by authors

Type II containers are assigned to Quay Crane 3. The QCs unload import containers to AGVs or retrieve export containers from AGVs according to the scheduling plan. If an AGV arrives at the QC later than the planned operation time, it can cause delays, potentially extending the vessel's stay at the berth and leading to cascading delays for multiple vessels. Additionally, mutual waiting between YCs and AGVs can easily result in further delays to the QC's scheduled operations.

Buffer brackets in automated terminals help reduce wait times between YCs and AGVs. AGV transport containers from QCs to buffer brackets, while YCs move them to handover areas or from export yards to buffer brackets for AGV collection. This process ensures continuous operation without mutual waiting. However, limited buffer brackets can become fully occupied by incoming containers if YCs wait for AGVs or if export containers occupy frames prematurely.

Classifying containers by attributes enhances their mobility on buffer brackets without compromising ship stability or increasing container re-handling during loading/unloading by QCs or YCs. Containers with identical attributes can have their working orders interchanged. The operational times of YCs vary depending on container positioning: lower bay numbers reduce trolly time in non-cantilever areas. Lower tier numbers decrease spreader time during container retrieval and smaller row differences between adjacent containers minimize spreader movement time. For specific export and import containers handled by AGVs, optimizing the sequence of handling of YC for same-type containers can reduce buffer bracket occupancy without disrupting QC schedules. As illustrated in Figure 2 , export containers are categorized into eight groups (Types I-VIII) based on destination port, size and weight. For instance, QC 4 is scheduled to load 14 Type II containers, identified as ①, ⑩, ⑪, ⑭, ⑰ and ⑳, located in Block 9. Container ① is situated in Bay 3, row 3, Tier 3, while container ⑩ is in Bay 1, row 6, Tier 3. When QC 4 requires a Type II container both containers ① and ⑩ are accessible. Prioritizing container

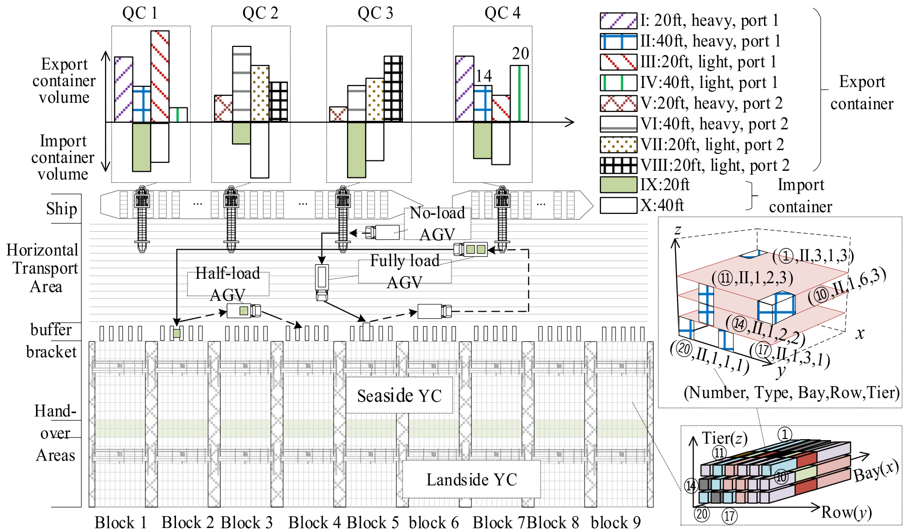

Figure 2. Flow of multi-attribute container transport for equipment in ACT. **Source(s):** Figure by authors

⑩ due to its shorter moving time can prevent QC delays if the YC is operating near the AGV's scheduled pick-up. Otherwise, container ① may be chosen to minimize buffer bracket occupation time while waiting for AGV arrival.

Sorting containers by attributes enhances the capacity of the YC to perform more double-cycle heavy-load operations. These operations entail the seaside YC picking up export containers from the handover area, placing them on buffer brackets and subsequently retrieving import containers from the buffer brackets to place them in the handover area. In non-cantilever blocks, YCs face operational span limitations, and their waiting time does not incur additional costs. Increasing these operations within a reasonable range can reduce movement expenses. The QC schedule dictates the latest time AGVs can collect export containers and the earliest time for delivering import containers. YC must position the designated export containers on buffer brackets before the AGV deadline and handover import containers to the handover area after AGV delivery. Traditionally, few combinations of import and export containers meet these time conditions. However, by classifying containers, YCs can optimize working time on export containers by selecting those from larger bays when time windows allow, thus increasing the number of viable double-cycle heavy-load operations.

The impact of various attributes on container type classification varies significantly. For export containers, factors such as destination port, size and weight relate to container turnaround time, cabin size and ship stability, respectively. It is inappropriate to exchange the handling sequence of containers with different attributes merely to reduce YC waiting times. Hence, export containers with varying destination ports, sizes and weights are considered distinct types for seaside cranes. Conversely, for import containers, the distance moved from the buffer brackets to the handover area remains constant, and the YC's operational time is minimally affected by attributes such as owner, size or weight. Therefore, import containers with different owners, sizes and weights are treated as the same type by seaside YC. The AGV's operational area lies between the QC and the block, with its capacity solely dependent on container size. An AGV can transport either two 20 ft containers or one 40 ft container simultaneously. For multi-load AGVs, all 20 ft import containers are classified as one type and all 40 ft import containers are classified as another. As depicted in Figure 2, import containers are categorized into two types (Types IX-X) based on size. An empty AGV picks up a 40 ft import container from QC 3 and transports it to block 5, placing it on a buffer bracket. The AGV then returns to QC 4, where it picks up two 20 ft import containers, delivers one to block 2 to place on a buffer bracket, and subsequently, in a half-load state, proceeds to block 4 to transport the other container.

Classifying containers by size can enhance the operational capacity of multi-load AGVs for heavy-load tasks. In the traditional model, multi-load AGV can transport either two consecutive 20 ft import or export containers or perform a sequence involving one import and one export container. As illustrated in Figure 3, if the initial task involves an import container received from the QC at point A, intended for placement in the block at point E, the movement paths for subsequent tasks of one-load and multi-load AGV comprise six scenarios. Among these, Cases 2, 4 and 6 provide two paths for multi-load AGV, indicating that their operational order can be interchanged at points E and F.

When AGVs handle two consecutive import containers, the heavy-load distance for multi-load AGVs increases compared to one-load operations, while the empty load path is shortened (Case 1–Case 4). Conversely, when handling one import and one export container consecutively, multi-load transport does not decrease transportation time and may even elevate full-load transport costs (Case 5–Case 6). Therefore, classifying two adjacent 20 ft containers as the same type by size can extend the heavy-load distance for AGVs. When AGVs manage import and export containers consecutively, they should prioritize delivering the import container to the yard first, and then, within the time window that accommodates the handover of the container under the QC, proceed to the blocks to handle the export container, ensuring a fully loaded round trip for the AGV.

3. Mathematical modeling

Based on the QC scheduling scheme, we know the number of containers of various types that YCs must pick up or place in each block during different time periods. To prevent any delays in the QC operations, a scheduling optimization model for YCs in each block is developed to

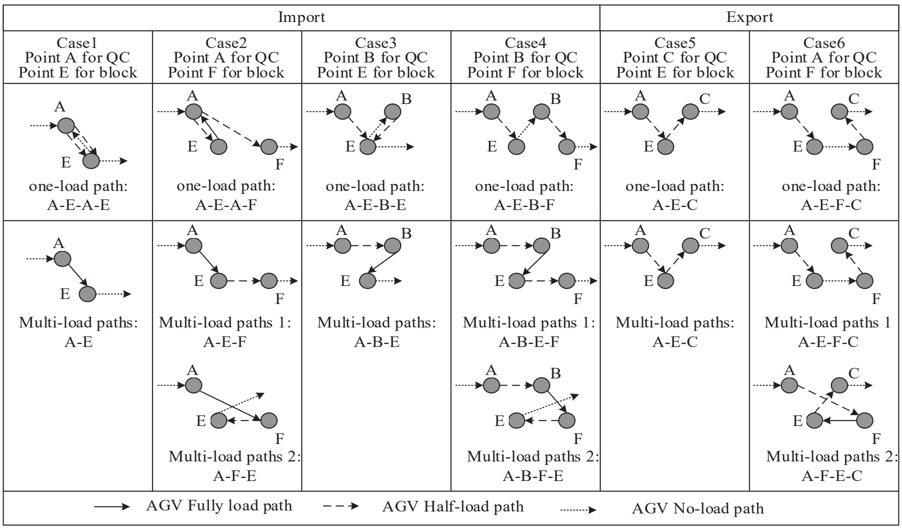

Figure 3. Scheduling scheme for one-load and multiple-load AGV. **Source(s):** Figure by authors

minimize their operating costs. Upon acquiring the YC operational scheme, the number of containers being loaded onto or unloaded from buffer brackets by YC in each block at any given moment is determined. Consequently, a dispatching optimization model is developed for all multi-load AGVs to minimize their operational costs.

3.1 Assumptions

(1) The seaside YC places the containers directly in the handover area upon arrival, without being affected by the capacity of the handover area or the landside YC.

(2) There are no containers from other ships that have not yet arrived in port above those awaiting loading and unloading.

(3) The operating efficiency of all YCs is the same, and a single YC can only handle one container at a time.

(4) The operating efficiency of all AGVs is the same, and each AGV can transport a maximum of one 40 ft container or two 20 ft containers simultaneously.

3.2 Description of symbols

Sets:

H	Set of blocks, $H = \{1, \ldots h, \ldots \vert H\vert\}$
R_h	Set of rows in block h, $R_h = \{1, \ldots r_h, \ldots \vert R_h\vert\}$
E_h	Set of rows tiers in block h, $E_h = \{1, \ldots e_h, \ldots \vert E_h\vert\}$, e_{nh} indicates the tier on which container n located
I	Set of container attribute types, $I = \{1, \ldots i, \ldots \vert I\vert\}$
T	Set of moments, $T = \{1, \ldots t, \ldots \vert T\vert\}$
N	Set of containers, $N = \{1, \ldots n, \ldots \vert N\vert\}$, $n \in N$, n_0 is a virtual container. N^+ indicates set of import containers, N^- indicates set of export containers. N_{f1} is set of 20 ft containers, N_{f2} is set of 40 ft containers. N_h is set of containers within block h, N_h^+ indicates the collection of import containers in the h, N_h^- indicates the collection of export containers in block h. N_{hr} is set of containers in r row of stack in block h. N_i is set of containers of type i. $\vert N_i^{t_d}\vert$ indicates the number of i-types that must be handled during period t_d. N_s is set of 20 ft containers that are transported by a multi-load AGV
K_a	Set of AGVs, $K_a = \{1, \ldots k_a, \ldots \vert K_a\vert\}$
K_q	Set of QCs, $K_q = \{1, \ldots k_q, \ldots \vert K_q\vert\}$
K_y	Set of YCs, $K_y = \{1, \ldots k_y, \ldots \vert K_y\vert\}$

Parameters:

Y_q	Buffer bracket capacity in a block
$T_{k_q n}$	The moment when the QC k_q operates container n
$T^0_{k_q n}$	The planned moment when container n handled by QC k_q is placed onto the AGV or picked up from the AGV
$T^1_{k_q n}$	The actual moment when container n handled by QC k_q is placed onto the AGV or picked up from the AGV
$T^0_{k_y n}$	The planned moment when YC k_y begins to handle container n
$T^1_{k_y n}$	The actual moment when YC k_y begins to handle container n
$T_{yk_a n}$	The moment when container n is placed onto the buffer brackets or is taken from the buffer brackets by AGV k_a
$t_{k_y nm}$	The time taken for YC k_y to go to the operating container m with no-load after it the end of handled container n

$t_{k_y n}$	The time taken for YC k_y to lift container n from its location and move it to the target location
$t_{k_a n}$	The time taken for AGV k_a to transport container n from its origin to its target location
t_d	The time node, is used to assist in determining whether the quantity of the same type of containers for loading and unloading meets the requirements
t_n^1	The time taken for the AGV to pick up/hand over a container under the QC
t_n^2	The time taken for the AGV to pick up/hand over a container at buffer brackets
t_{nm}^1	The time for the AGV to reach the starting point of container m after handling over the container from the end point of container n
t_{nm}^2	The time taken by the AGV in the half-load state to transport containers n and m during one round is the shortest
t_{nm}^3	The time taken by the AGV in the fully load state to transport containers n and m simultaneously is the shortest
$t_{qk_a n}^w$	The time taken by AGV k_a waiting under the QC during the transport of container n
$t_{yk_a n}^w$	The time taken by AGV k_a waiting under the YC during the transport of container n
c_1	The cost per unit time of no-load movement for a YC
c_2	The cost per unit time of half-load movement for an AGV
c_3	The cost per unit time of full-load movement for an AGV
c_4	The cost per unit time of waiting for an AGV
c_5	The cost per unit time of no-load movement for an AGV

Decision variables:

$x_{k_y n}^t$	1, if YC k_y handle container n during period t; 0, otherwise
$x_{k_y nm}$	1, if YC k_y handle container m after complete container n; 0, otherwise
$p_{k_y}^t$	The number of containers on the buffer brackets in period t
$y_{k_a n}$	1, if k_a transports container n; 0, otherwise
$y_{k_a nm}$	1, if k_a transports container m after container n; 0, otherwise
z_{nm}	1, if an AGV goes to pick up container m with taking n; 0, otherwise

3.3 Mixed integer programming model-phase 1

$$\min f_1 = c_1 \cdot \sum_{n \in N_h} \sum_{m \in N_h} x_{k_y nm} t_{k_y nm} \tag{1}$$

s.t.

$$\sum_{t \in T} x_{k_y n}^t = 1, \forall n \in N_h \tag{2}$$

$$\sum_{n \in N_h} x_{k_y n}^t \leq 1, \forall t \in T \tag{3}$$

$$\sum_{n \in N_h} \sum_{t \in T} x_{k_y n}^t = |N_h| \tag{4}$$

$$\sum_{n \in N_h \setminus \{m\}} x_{k_y nm} = 1, \forall m \in N_h \cup \{n_0\} \tag{5}$$

$$\sum_{m \in N_h \setminus \{n\}} x_{k_y nm} = 1, n \in N_h \cup \{n_0\} \tag{6}$$

$$u_m \geq u_n + |N_h| \times \left(x_{k_y nm} - 1\right) + 1, \forall n, m \in N_h, n \neq m \tag{7}$$

$$\sum_{t \in T} \left(t \times x^t_{k_y m}\right) + |T| \geq \sum_{t \in T} \left(t \times x^t_{k_y n}\right) + |T| \times x_{k_y nm} + t_{k_y n} + t_{k_y nm}, \forall n \in N_h, m \in N_h \tag{8}$$

$$\sum_{t \in T} \left(t \times x^t_{k_y m}\right) \geq \sum_{t \in T} \left(t \times x^t_{k_y n}\right) + t_{k_y n}, \forall n \in N_{hr}, m \in N_{hr}, e_{hn} \geq e_{hm} \tag{9}$$

$$0 \leq p^t_{k_y} \leq Y_q, \forall t \in T \tag{10}$$

$$p^t_{k_y} + Y_q \times x^t_{k_y n} \leq p^{t-1}_{k_y} + Y_q - 1, \forall t \in T, n \in N^+_h \tag{11}$$

$$p^t_{k_y} + Y_q \times x^{t-t'}_{k_y n} \leq p^{t-1}_{k_y} + Y_q + 1, \forall n \in N^-_h, t \in T, t' \in \left[t_{k_y n} - t_{nm}, t_{k_y n} - t_{nm} + 1\right) \tag{12}$$

$$p^t_{k_y} \leq p^{t-1}_{k_y} + 1, \forall n \in N^+_h, t \in \left[T^1_{k_q n} + t_{k_a n}, T^1_{k_q n} + t_{k_a n} + 1\right) \tag{13}$$

$$p^t_{k_y} \leq p^{t-1}_{k_y} - 1, \forall n \in N^-_h, t \in \left[T^1_{k_q n} - t_{k_a n}, T^1_{k_q n} - t_{k_a n} + 1\right) \tag{14}$$

$$\sum_{t \in T} \left(t \times x^t_{k_y n}\right) \geq T^1_{k_q n} + t_{k_a n}, \forall n \in N^+_h \tag{15}$$

$$\sum_{t \in t_d} x^t_{k_y n} \geq |N^{t_d}_i|, \forall n \in N^-_h \cap N_i, t_d \in T, i \in I \tag{16}$$

$$u_n \in R+, \forall n \in N_h \tag{17}$$

$$x^t_{k_y n} \in \{0, 1\}, x_{k_y nm} \in \{0, 1\}, \forall n, m \in N_h \tag{18}$$

The objective function (1) is designed to minimize the cost associated with the infield crane's empty travel operations. Constraint (2) requires each container to be handled once. Constraint (3) limits a YC to handle at most one container at a time. Constraint (4) ensures that all containers are handled. Constraint (5) indicates that each container has at most one immediate predecessor operation, where n_0 represents the virtual container and the distance between any container and the virtual container is zero. Constraint (6) indicates that each container has at most one immediate successor operation. Constraint (7) uses an auxiliary variable u_n to avoid subtour. Constraint (8) dictates the movement of a YC between adjacent containers. Constraint (9) prioritizes upper tier containers for handling. Constraint (10) limits buffer bracket capacity. Constraints (11) and (12) are induced by YC operations. Constraints (13) and (14) are caused by AGV operations. Constraint (15) specifies the time window for the export containers handled by the YC. Constraint (16) specifies the quantity for the same type of export containers that must be processed by the YC. The left side of the constraint represents the quantity of i-type containers handled by gantry crane k_y before time td, while the right side represents the number of *i*-type containers that must be handled during period t_d, which is an input value derived from the overall workload schedule. Constraint (17) confirms u_n is a positive real number. Constraint (18) defines decision variable ranges.

3.4 Mixed integer programming model-phase 2

$$\min f_2 = \sum_{n\in N_{f_1}}\sum_{m\in N_{f_1}}\sum_{k_a\in K_a} z_{nm}y_{k_anm}\left(c_3t_{nm}^2 + c_4t_{nm}^3\right) + \sum_{n\in N_{f_1}}\sum_{m\in N}\sum_{k_a\in K_a}(1-z_{nm}) \left(c_2y_{k_anm}t_{nm}^1 + c_3\left(1-\sum_{m\in N}z_{nm}y_{k_anm}\right)y_{k_an}t_{k_an}\right) + \sum_{n\in N_{f_2}}\sum_{m\in N}\sum_{k_a\in K_a}(1-z_{nm}) \left(c_2y_{k_anm}t_{nm}^1 + c_4\left(1-\sum_{m\in N}z_{nm}y_{k_anm}\right)y_{k_an}t_{k_an}\right) + \sum_{n\in N}\sum_{k_a\in K_a}c_5\left(t_{qk_an}^w + t_{yk_an}^w\right) \quad (19)$$

s.t.

$$\sum_{k_a\in K_a} y_{k_an} = 1, \forall n\in N/\{n_0\} \quad (20)$$

$$\sum_{m\in N} y_{k_anm} - \sum_{m\in N} y_{k_amn} = 1, \forall n\in N, k_a\in K_a \quad (21)$$

$$\sum_{n\in N_{f_1}} z_{nm} \leq 1, \forall m\in N_{f_1} \quad (22)$$

$$\sum_{n\in N} z_{nm} + \sum_{n\in N} z_{mn} = 0, \forall m\in N_{f_2} \quad (23)$$

$$p_{k_y}^t \leq p_{k_y}^{t-1} + 1, \forall n\in N^+, t\in \left[T_{yk_an}, T_{yk_an}+1\right), k_y\in K_y \quad (24)$$

$$p_{k_y}^t \leq p_{k_y}^{t-1} - 1, \forall n\in N^-, t\in \left[T_{yk_an}, T_{yk_an}+1\right), k_y\in K_y \quad (25)$$

$$p_{k_y}^t \leq p_{k_y}^{t-1} - 1, \forall n\in N^+, t\in \left[T_{k_yn}^1, T_{k_yn}^1+1\right), k_y\in K_y \quad (26)$$

$$p_{k_y}^t \leq p_{k_y}^{t-1} + 1, \forall n\in N^-, t\in \left[T_{k_yn}^1, T_{k_yn}^1+1\right), k_y\in K_y \quad (27)$$

$$p_{k_y}^t \leq Y_q, \forall t\in T, k_y\in K_y \quad (28)$$

$$T_{yk_an} \geq T_{k_qn}^1 + t_h^1 + t_{k_an} + t_h^2 + t_{yk_an}^w, \forall n\in N^+ \quad (29)$$

$$T_{yk_am} = \max\left(y_{k_anm}\left(T_{yk_an} + t_{nm}^1\right), T_{k_ym}^1 + t_{k_ym}\right), \forall n\in N^+, m\in N^- \quad (30)$$

$$T_{yk_am} = \max\left(y_{k_anm}\left(T^1_{k_qn} + t^1_n + t^1_{nm}\right), T^1_{k_ym} + t_{ym}\right), \forall n, m \in N^- \tag{31}$$

$$t^w_{yk_an} \leq T^1_{k_yn} + t_{k_yn} - T_{yk_an}, \forall n \in N^- \tag{32}$$

$$T^1_{k_qm} \geq \max\left(T^0_{k_qm}, \left(T_{yk_am} + t^2_n + t_{k_am}\right)\right), \forall m \in N^+ \tag{33}$$

$$T^1_{k_qm} \geq \max\left(T^0_{k_qm}, y_{k_anm}\left(T^1_{k_qn} + t^1_{nm}\right)\right), \forall n \in N^+, m \in N^- \tag{34}$$

$$T^1_{k_qm} \geq \max\left(T^0_{k_qm}, y_{k_anm}(1 - z_{nm})\left(T_{yk_an} + t^2_n + t_{k_an}\right)\right), \forall n, m \in N^-_h \tag{35}$$

$$T^1_{k_qm} \geq \max\left(T^0_{k_qm}, y_{k_ann'}y_{k_an'm}z_{nn'}(1 - z_{n'm})\left(T_{yk_an} + 2t^1_n + 2t^2_n + t^2_{nn'} + t^3_{nn'} + t^1_{n'm}\right)\right), \\ \forall n, n', m \in N^+, (n, n') \in N_s \tag{36}$$

$$T^1_{k_qm} \geq \max\left(T^0_{k_qm}, y_{k_ann'}y_{k_an'm}z_{nn'}(1 - z_{n'm})\left(T_{yk_an} + 2t^1_n + 2t^2_n + t^2_{nm} + t^3_{nm}\right)\right), \\ \forall n, n', m \in N^-, (n, n') \in N_s \tag{37}$$

$$t^w_{qk_an} \leq \max\left(T^0_{k_qn} - T^1_{k_qn}, 0\right), \forall n \in N \tag{38}$$

$$T^1_{k_qm} = \max\left(T^1_{k_qn} - T^0_{k_qn}, 0\right) + T^0_{k_qm}, \forall n, m \in N, m > n, k_q \in K_q \tag{39}$$

$$T^1_{k_ym} = \max\left(T^1_{k_yn} + t_{k_yn} + t_{nm}, T^0_{k_ym}\right), \forall n, m \in N, m > n, k_y \in K_y \tag{40}$$

$$\max_{n \in N} T^1_{k_qn} \leq \max_{n \in N} T^0_{k_qn} \tag{41}$$

$$y_{k_an} \in \{0, 1\}, y_{k_anm} \in \{0, 1\}, z_{nm} \in \{0, 1\}, \forall n, m \in N, k_a \in K_a \tag{42}$$

The objective function (19) represents the minimization of the transportation cost of the AGV. Constraint (20) means that each container can only be transported by one AGV. Constraint (21) indicates that each container has one and only one immediate predecessor and successor operation. Constraint (22) means that, only one 20 ft container can be transported with another 20 ft container by the same AGV at the same time. Constraint (23) means that all 40 ft containers cannot be transported with other containers by the same AGV at the same time. Constraints (24)–(27) represent the rules for updating the number of containers on the buffer brackets. Constraint (28) indicates the capacity limits of the buffer brackets. Constraint (29) represents the time conditions that must be met for the AGV to hand over imported containers to the YC. Constraints (30) and (31) respectively indicate the moment for the AGV to pick up export container m, while when the predecessor container n is the import container and the export container, the AGV picks up export container m from the buffer brackets. Constraint (32) indicates the waiting time in the block when the AGV picks up export container *n*. Constraints (33)–(37) represent the actual time when QC k_q operates on container *m*. Constraint (33) is for import container *m*, while constraints (34) and (35) are for export container *m*. Constraint (34) is for the case where the preceding container *n* is an import container, and constraint (35) is for the case where the preceding container *n* is an export container. Constraint (36) is also for import container *m*, but specifically when its preceding task is a pair of 20 ft containers *n* and *n'*. Compared to constraint (36), constraint (37) differs in that container m is an export container. Constraint (39) indicates the operation time of updating the subsequent QC scheme after the delay. Constraint (40) indicates the time for updating the subsequent YC's plan after the delay. Constraint (41) indicates that the total completion time of QC operations must not be delayed. Constraint (42) indicates the range of values of the decision variable.

4. Hybrid exact-heuristic algorithms

The GUROBI solver can directly solve the first phase of the model. The multi-load AGV scheduling problem for transporting import containers has been proven by Kong *et al*. (2024) to be NP-complete, with computation time growing exponentially as the problem size increases. This paper builds on the research by Kong *et al*. (2024) by extending it to include an analysis of export containers, which results in the problem also being NP-complete. Therefore, a two-stage approach is more suitable for solving the model.

The scheming period *P* is segmented into equal parts, with GUROBI solving YC schedules within each segment. Subsequently, a variable neighborhood search immune algorithm (VNS-IA) addresses AGV scheduling and updating schemes based on segment completion and delays.

4.1 Coding and decoding

Decimal encoding is used to represent the sequence of AGV transport containers. As depicted in Figure 4, the 10 containers are categorized into two groups based on their varying attribute values. During the generation and mutation of antibodies, AGVs are randomly assigned container numbers to ensure compliance with constraints (20)–(23). An AGV can carry full load if it sequentially handles two 20 ft export or import containers without QC delays. Otherwise, it operates with half-load. The antibody's fitness value is set as the inverse of the objective function value.

4.2 Neighborhood structure

Four structures, as shown in Figure 5, are used to enhance the search depth of the algorithm.

(1) Insertion: randomly select a container, insert it behind another container at a random position, and have it transported by the same AGV. For example, after container 6 in Figure 5(a) is inserted behind container 5, it is transported by AGV 1.

(2) Swap: randomly select two positions and swap the order of the containers at those positions, but keep the AGV allocation scheme unchanged. As shown in Figure 5(b), five containers, including container 5 and container 3, have exchanged positions.

(3) Local Perturbation: randomly determine the number of containers each AGV needs to transport, but keep the order of container operations unchanged. As shown in Figure 5(c), the workload of AGV 1 and AGV 3 is reduced, while that of AGV 2 is increased.

Container	1	4	5	2	6	10	3	7	8	9
AGV	1	1	1	2	2	2	3	3	3	3
Type	2	1	1	2	2	2	1	1	1	2
QC	1	2	2	2	1	1	2	1	1	2
YC	6	2	2	1	1	5	4	3	2	6

Figure 4. Schematic diagram of antibody structure. **Source(s):** Figure by authors

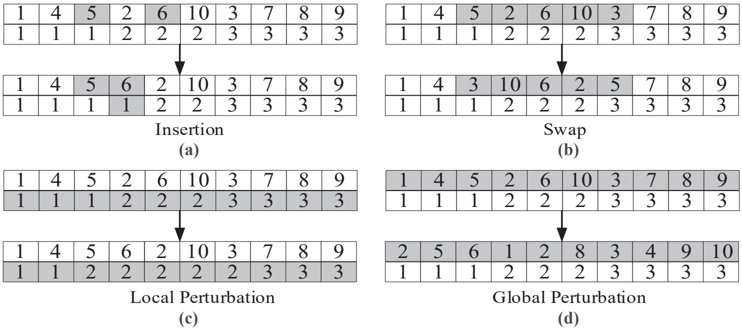

Figure 5. Schematic diagram of the neighborhood structures. **Source(s):** Figure by authors

(4) Global Perturbation: randomly determine the order of container operations but keep the number of containers each AGV handles constant. As shown in Figure 5(d), the workload of the three AGVs remains unchanged, but the containers they handled have all changed.

4.3 Adjusting infeasible solutions

During the decoding process, AGVs may fail to pick up/deliver containers at the cranes on time due to the unreasonable order of individual containers, leading to crane delays of duration Δt. If the current phase is not the last cycle of the rolling loop, let c_6 be the penalty cost per unit time of crane delay, denoted as fitness $F = 1 / (f_2 + c_6 \Delta t)$. When solving the scheduling scheme within the last cycle, antibodies that do not meet the contracted laytime are assigned a fitness of 0 and are not passed on to the next generation.

4.4 Termination conditions

When the iteration count reaches its limit, the algorithm terminates.

5. Numerical experiments

We designed a series of numerical experiments based on the layout and equipment operation parameters of Yangshan Ⅳ ACT of Shanghai Port. The relevant input parameters are shown in Table 1. All experiments are coded in MATLAB and the optimization solver used is GUROBI 9.5.2. The calculation environment was a PC with an Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz and 8 GB of RAM.

5.1 Small-scale experiments

To verify the correctness of the algorithm, the stealth enumeration algorithm (SEA) is used to solve small-scale instances and compare the results with those obtained using the improved VNS-IA. The SEA involves enumerating all feasible paths and directly solving the AGV

Table 1. Input parameters

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
t_h^1	60s	c_1	150*yuan/h*	c_3	40*yuan/h*	c_5	20*yuan/h*
t_h^2	6s	c_2	30*yuan/h*	c_4	45*yuan/h*	c_6	10000*yuan/h*

Source(s): Own elaboration

scheduling model. The improved VNS-IA runs each instance 10 times and calculates the average value. The comparison results are shown in Table 2.

Table 2 illustrates that AGV costs vary depending on the case characteristics. Scenarios with a higher number of containers exhibit higher AGV costs. When the number of containers and AGVs exceeds a certain threshold, enumerating all feasible AGV schedules becomes infeasible due to the extensive solution space. VNS-IA addresses more cases than the SEA does. When comparing the Gap1 values, it is evident that although VNS-IA might not achieve the optimal solution for large numbers of containers, the performance gap remains small. In terms of runtimes, the efficiency of this algorithm is comparable to that of the exact method.

5.2 Large-scale experiments

To validate the superiority of VNS-IA over other heuristic algorithms, we designed and solved large-scale test cases using the immune algorithm, the local search algorithm and the algorithm from this paper to calculate the objective function values. Under the condition of the same number of iterations, each algorithm was executed 10 times and the average cost and run time values were statistically analyzed, as shown in Table 3.

Table 3 demonstrates that all the two heuristic algorithms consistently produce satisfactory solutions across the 23 scenarios examined. When analyzing the “Gap2” column, it reveals that while the VNS can sometimes closely approximate the optimal solution, it often fails to escape from local optima, thus exhibiting instability. Comparing the “Gap3” column shows that although the solution time of VNS-IA is relatively slower, the minimum objective function values obtained are lower than those of the IA, reflecting the IA’s inability to conduct in-depth searches. The VNS-IA combines the advantages of the two algorithms, taking longer in the solution process than both, yet offering superior solution quality and efficiency.

Table 2. Comparison between the SEA and VNS-IA

No.	$N/k_q/k_y/k_a$	SEA f_2/yuan	SEA Runtime/s	VNS-IA f_2/yuan	VNS-IA Runtime/s	*Gap*1 (%)
1	10/1/3/3	5.56	2.15	5.56	7.94	0.00
2	10/1/3/4	5.01	0.98	5.01	7.13	0.00
3	10/1/3/5	3.59	0.49	3.59	8.79	0.00
4	10/1/3/6	3.61	0.14	3.61	6.56	0.00
5	10/2/6/6	5.32	0.11	5.32	4.74	0.00
6	10/2/6/7	5.53	0.10	5.53	4.75	0.00
7	10/2/6/8	3.25	0.08	3.25	3.84	0.00
8	10/2/6/9	3.17	0.02	3.17	0.00	0.00
9	15/1/3/3	–	–	10.16	12.55	–
10	15/1/3/4	–	–	7.40	11.55	–
11	15/1/3/5	–	–	5.85	11.26	–
12	15/1/3/6	5.19	7024.97	5.59	6.85	−7.16
13	15/2/6/6	–	–	8.62	4.83	–
14	15/2/6/7	9.05	34630.42	9.28	3.57	−2.48
15	15/2/6/8	7.85	5002.42	8.40	6.60	−6.55
16	15/2/6/9	7.87	640.28	8.33	5.89	−5.52
17	20/2/6/6	–	–	10.89	8.05	–
18	20/2/6/7	–	–	12.62	7.63	–
19	20/2/6/8	–	–	12.87	7.33	–
20	20/2/6/9	–	–	11.93	5.53	–

Note(s): $Gap1 = \frac{f_{SEA}-f_{VNS-IA}}{f_{VNS-IA}} \times 100\,\%$. ”—” means the number of solutions exceeds the computational space
Source(s): Own elaboration

Table 3. Comparison of the results of the VNS and IA with VNS-IA

No.	$N/k_q/k_y/k_a$	VNS f_2/yuan	Runtime/s	IA f_2/yuan	Runtime/s	VNS-IA f_2/yuan	Runtime/s	*Gap*2 (%)	*Gap*3 (%)
21	100/2/6/7	99.2	7.3	142.5	10.6	86.0	27.4	15.3	65.6
22	100/2/6/8	85.8	7.9	157.4	10.3	81.6	24.6	5.1	92.9
23	100/2/6/9	112.2	8.0	154.2	10.0	91.2	20.6	23.1	69.2
24	100/2/6/10	86.7	7.8	158.8	9.5	86.0	26.6	0.8	84.6
25	200/2/6/7	483.1	73.0	814.9	43.3	473.1	110.9	2.1	72.3
26	200/2/6/8	515.7	65.3	959.1	42.0	510.6	101.4	0.9	84.2
27	200/2/6/9	508.0	78.3	1015.9	43.3	500.4	102.0	1.5	99.1
28	200/2/6/10	548.3	67.9	844.2	42.5	469.6	104.7	16.8	79.8
29	200/3/9/11	588.8	71.5	979.9	41.1	496.5	93.6	18.6	97.3
30	200/3/9/12	572.6	75.6	1041.5	42.2	530.5	93.9	7.9	96.3
31	200/3/9/13	482.7	78.3	908.0	41.0	480.4	95.2	0.5	85.0
32	200/3/9/14	532.4	76.4	910.1	39.4	472.9	98.7	12.6	92.5
33	200/3/9/15	597.9	73.7	949.2	37.4	509.7	100.5	17.3	86.2
34	200/4/12/16	251.8	41.7	396.9	29.3	248.9	72.1	1.1	59.4
35	200/4/12/17	253.2	40.0	394.7	25.9	249.8	71.6	1.3	58.0
36	200/4/12/18	237.7	39.5	394.1	26.0	226.2	77.6	5.1	74.3
37	200/4/12/19	246.5	43.9	388.8	27.2	229.5	71.5	7.4	69.4
38	200/4/12/20	234.8	44.2	398.4	26.1	228.5	65.5	2.8	74.4
39	300/4/12/16	696.4	165.3	1052.4	59.8	662.5	190.1	5.1	61.5
40	300/4/12/17	717.8	174.2	1078.7	64.9	618.0	176.7	16.2	74.6
41	300/4/12/18	776.0	171.8	1066.6	63.6	722.8	179.2	7.4	47.6
42	300/4/12/19	685.8	172.3	1049.8	61.0	588.9	174.1	16.5	78.3
43	300/4/12/20	623.4	193.6	1027.4	61.2	602.8	182.2	3.4	69.2

Note(s): $Gap2 = \frac{f_{VNS} - f_{VNS-IA}}{f_{VNS-IA}} \times 100\ \%$ $Gap3 = \frac{f_{IA} - f_{VNS-IA}}{f_{VNS-IA}} \times 100\ \%$
Source(s): Own elaboration

5.3 Sensitivity analysis

This section demonstrates the cost-effectiveness of the proposed container batch grouping strategy through comparative experiments. First, 24 comparative scenarios were designed with varying proportions of 40-ft containers (0%, 20%, 40%, 60%, 80% and 100%) and unloading container ratios (25%, 50% and 75%). Subsequently, the model and algorithms proposed in this paper were applied to calculate the operational costs of AGVs and YCs for handling 150 containers using 9 YCs and 14 AGVs. Finally, under the same scenarios, Constraint (16) was relaxed to allow YCs to operate containers under a first-come-first-served (FCFS) principle and their operational costs were computed. The comparison results are shown in Figure 6.

From Figure 6, it can be observed that as the number of container increases, the costs associated with AGVs and YCs gradually rise. However, in the same case study, when applying the container batch grouping strategy, AGV costs are consistently lower than when not considering this attribute, as shown in the left subplot of Figure 6. In most cases, the YC costs are also lower when using the container batch grouping strategy compared to not considering this attribute. However, in certain scenarios, the YC costs may be higher when applying the container batch grouping strategy, as illustrated in the right subplot of Figure 6. This is because, as the proportion of unloaded 40 ft containers increases, the container batch grouping strategy may lead to insufficient buffer rack capacity, resulting in increased waiting costs for YC. Therefore, the container batch grouping strategy should be dynamically adjusted based on the proportion of different container types in the specific scenario.

6. Conclusions

This study focuses on the joint scheduling problem of YCs and multi-load AGVs in an ACT. The main conclusions are as follows:

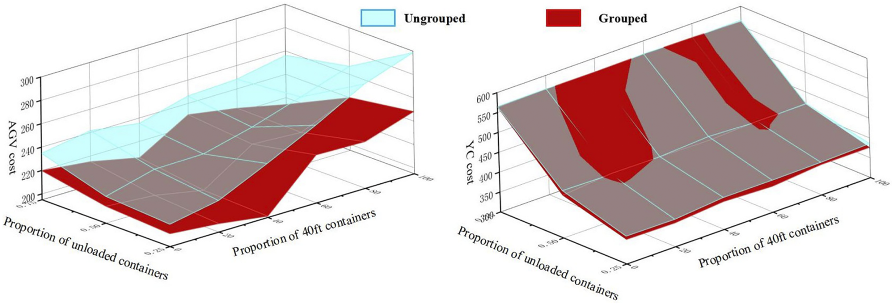

Figure 6. Operating costs of AGVs and yard cranes with and without considering grouping. **Source(s):** Figure by authors

(1) Considering the multiple attributes of containers, the payload capacity of AGVs, and the buffer bracket capacity in each block, mixed integer programming models are developed for YC scheduling and AGV scheduling.

(2) The YC scheduling model is directly solved using GUROBI optimization software to verify its correctness. A VNS-IA is designed to solve the AGV scheduling model. Experimental results show that the algorithm presented in this paper has higher solution quality and greater efficiency.

(3) By classifying containers based on their multiple attributes and optimizing the handling sequence of the same-type containers, the ACT can effectively achieve a reduction in the total cost of up to 8.9%.

From a management perspective, considering container classification and interchangeability in this study can lead to more flexible and efficient terminal operations. Thoughtful container classification enables a more precise planning of equipment, pathways and resources, consequently enhancing overall port efficiency. Substituting containers contribute to a partial balance in resource allocation and operational workflows, which effectively addresses unexpected situations and changes during the cargo handling process, ultimately improving the adaptability and resilience of the terminal operations system.

In the future, research directions could focus on optimizing container classification and interchangeability strategies to enhance terminal operations. One potential area is developing dynamic container classification systems that consider real-time data, such as container size, weight and destination, to improve resource allocation and operational efficiency. Another direction is exploring advanced interchangeability algorithms that prioritize container compatibility and minimize disruptions during cargo handling. Additionally, integrating machine learning and IoT technologies could enable predictive modeling of container flow and resource demand, further enhancing adaptability. Finally, research could investigate the environmental impact of container classification and interchangeability, aiming to develop sustainable practices that reduce energy consumption and emissions while maintaining operational efficiency.

References

Chen, X., He, S., Zhang, Y., Tong, L.C., Shang, P. and Zhou, X. (2020), “Yard crane and AGV scheduling in automated container terminal: a multi-robot task allocation framework”, *Transportation Research Part C: Emerging Technologies*, Vol. 114, pp. 241-271, doi: 10.1016/j.trc.2020.02.012.

Chen, S., Wang, H. and Meng, Q. (2021), "Autonomous truck scheduling for container transshipment between two seaport terminals considering platooning and speed optimization", *Transportation Research Part B: Methodological*, Vol. 154, pp. 289-315, doi: 10.1016/j.trb.2021.10.014.

Dragović, B., Zrnić, N., Tzannatos, E., Kosanić, N. and Dragović, A. (2023), "A bibliometric analysis and assessment of container terminal operations research", *Maritime Business Review*, Vol. 8 No. 3, pp. 269-293, doi: 10.1108/mabr-07-2022-0035.

Hsu, H.P., Tai, H.H., Wang, C.N. and Chou, C.C. (2021), "Scheduling of collaborative operations of yard cranes and yard trucks for export containers using hybrid approaches", *Advanced Engineering Informatics*, Vol. 48, 101292, doi: 10.1016/j.aei.2021.101292.

Kong, L., Ji, M., Yu, A. and Gao, Z. (2024), "Scheduling of automated guided vehicles for tandem quay cranes in automated container terminals", *Computers and Operations Research*, Vol. 163, 106505, doi: 10.1016/j.cor.2023.106505.

Lam, J.S.L. and Chen, Z.S. (2021), "Survey of economic, energy and environmental aspects of cargo handling equipment in container ports", *2021 6th International Conference on Transportation Information and Safety (ICTIS)*, IEEE, pp. 1357-1363.

Naeem, D., Gheith, M. and Eltawil, A. (2023), "A comprehensive review and directions for future research on the integrated scheduling of quay cranes and automated guided vehicles and yard cranes in automated container terminals", *Computers and Industrial Engineering*, Vol. 179, 109149, doi: 10.1016/j.cie.2023.109149.

Oladugba, A.O., Gheith, M. and Eltawil, A. (2023), "A new solution approach for the twin yard crane scheduling problem in automated container terminals", *Advanced Engineering Informatics*, Vol. 57, 102015, doi: 10.1016/j.aei.2023.102015.

Tao, Y., Zhang, S., Lin, C. and Lai, X. (2023), "A bi-objective optimization for integrated truck operation and storage allocation considering traffic congestion in container terminals", *Ocean and Coastal Management*, Vol. 232, 106417, doi: 10.1016/j.ocecoaman.2022.106417.

Vallada, E., Belenguer, J.M., Villa, F. and Alvarez-Valdes, R. (2023), "Models and algorithms for a yard crane scheduling problem in container ports", *European Journal of Operational Research*, Vol. 309 No. 2, pp. 910-924, doi: 10.1016/j.ejor.2023.01.047.

Wang, Z. and Zeng, Q. (2022), "A branch-and-bound approach for AGV dispatching and routing problems in automated container terminals", *Computers and Industrial Engineering*, Vol. 166, 107968, doi: 10.1016/j.cie.2022.107968.

Xing, Z., Liu, H., Wang, T., Chew, E.P., Lee, L.H. and Tan, K.C. (2023), "Integrated automated guided vehicle dispatching and equipment scheduling with speed optimization", *Transportation Research Part E: Logistics and Transportation Review*, Vol. 169, 102993, doi: 10.1016/j.tre.2022.102993.

Xu, B., Jie, D., Li, J., Yang, Y., Wen, F. and Song, H. (2021), "Integrated scheduling optimization of U-shaped automated container terminal under loading and unloading mode", *Computers and Industrial Engineering*, Vol. 162, 107695, doi: 10.1016/j.cie.2021.107695.

Yue, L.J., Fan, H.M. and Fan, H. (2023), "Blocks allocation and handling equipment scheduling in automatic container terminals", *Transportation Research Part C: Emerging Technologies*, Vol. 153, 104228, doi: 10.1016/j.trc.2023.104228.

Zhong, M., Yang, Y., Dessouky, Y. and Postolache, O. (2020), "Multi-AGV scheduling for conflict-free path planning in automated container terminals", *Computers and Industrial Engineering*, Vol. 142, 106371, doi: 10.1016/j.cie.2020.106371.

Corresponding author
Jasmine Siu Lee Lam can be contacted at: jasmlam@dtu.dk