Hindawi
Scientific Programming
Volume 2020, Article ID 8863994, 11 pages
https://doi.org/10.1155/2020/8863994

Research Article

Optimization of Continuous Berth Scheduling by Taking into Account Double-Line Ship Mooring

Cheng Luo, Hongying Fei, Dana Sailike, Tingyi Xu, and Fuzhi Huang

School of Management, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Hongying Fei; feihy@shu.edu.cn

Received 2 July 2020; Revised 20 August 2020; Accepted 31 August 2020; Published 18 September 2020

Academic Editor: Tingsong Wang

"Double-Line Ship Mooring" (DLSM) mode has been applied as an initiative operation mode for solving berth allocation problems (BAP) in certain giant container terminals in China. In this study, a continuous berth scheduling problem with the DLSM model is illustrated and solved with exact and heuristic methods with an objective to minimize the total operation cost, including both the additional transportation cost for vessels not located at their minimum-cost berthing position and the penalties for vessels not being able to leave as planned. First of all, this problem is formulated as a mixed-integer programming model and solved by the CPLEX solver for small-size instances. Afterwards, a particle swarm optimization (PSO) algorithm is developed to obtain good quality solutions within reasonable execution time for large-scale problems. Experimental results show that DLSM mode can not only greatly reduce the total operation cost but also significantly improve the efficiency of berth scheduling in comparison with the widely used single-line ship mooring (SLSM) mode. The comparison made between the results obtained by the proposed PSO algorithm and that obtained by the CPLEX solver for both small-size and large-scale instances are also quite encouraging. To sum up, this study can not only validate the effectiveness of DLSM mode for heavy-loaded ports but also provide a powerful decision support tool for the port operators to make good quality berth schedules with the DLSM mode.

1. Introduction

With the deepening of economic globalization, the world container transportation volume has increased dramatically in recent years [1]. As one of the most important components of container transportation network, container terminals play an important role in world economy. Considering that the efficiency of berth allocation problem has great impact on the output of container terminals, a lot of studies have been dedicated to the berth scheduling problems.

In general, berth schedules are determined by specifying berthing time and position for the coming vessels by taking into account various constraints, such as the berth capacity, announced arrival time and departure time of container ships, and certain specific berthing requirements. In order to avoid collisions between vessels, the single-line ship mooring (SLSM) mode, which specifies that "no more than one vessel can be allocated to the same berth position at the same time," is normally applied in container terminals over the world, and this rule is regarded as a default in most of the studies on berth scheduling [2].

As one of the most important economic role in the world economy, China is continuously developing the economic innovation, resulting in huge container throughput of the international hubs in China. Yangshan deep-water port is located in Shanghai, a mega city in China. As the largest sea-island artificial deep-water port, Yangshan deep-water port is an important part of Shanghai International Shipping Center, and its annual throughput has increased constantly since it was built in the year of 2005.

Having been one of the world's busiest container terminals, Yangshan deep-water port has applied a so-called "Double-Line Ship Mooring" (DLSM) mode to build berth schedules since the year of 2019. Different from the widely used SLSM mode, DLSM mode allows two container ships to be moored simultaneously at the same berth location, enabling more container ships to be moored at their ideal berth and cranes to operate two container ships at the same time, a reasonable way to improve the efficiency of berth operation. Despite that, the real efficiency of the port with DLSM mode depends on how sophisticated the operators are according to the investigation made at Yangshan port, because berth scheduling with DLSM is much more complex than with the SLSM system. In consequence, it is essential to develop an effective decision support system to help berth operators improve the quality and efficiency of daily schedules with the DLSM mode.

This study aims at minimizing the additional operation costs for the vessels not placed at their ideal berthing position and penalties for the ships not being able to leave before their preplanned departure time, for a continuous berth allocation problems (BAP) model with the DLSM mode. The main contributions of this study are as follows: (1) as the first study on berth scheduling with the DLSM mode, it validates the contribution of DLSM model in comparison with the widely used SLSM mode; (2) a mixed integer programming model is constructed for the continuous BAP with DLSM mode, enabling a benchmark for the future studies on such topic; (3) a PSO algorithm is developed to obtain good quality DLSM berthing schedules within reasonable execution time, offering the berth operators at busy ports a powerful decision support tool to improve their work efficiency.

The reminder of this paper is structured as follows. Section 2 is dedicated to the literature review of related work. The problem description is given in Section 3. In Section 4, the targeted problem is formulated as a mixed integer programming model, which can be solved by CPLEX solver, and then, the details of the proposed PSO algorithm are given in Section 5. Section 6 is dedicated to discussions about the experimental results, and this paper ends up with conclusions and perspectives.

2. Related Work

According to the literature, berth allocation problem (BAP) is regarded as the most important issue faced by the management of container terminals. As the quality of berth schedule has a great influence on the improvement of port operation efficiency, a lot of researchers have been studying BAPs, and numerous results were published [2]. Since Imai et al. [3] firstly addressed a static berth allocation problem (SBAP) and further developed a dynamic BAP model (DBAP for public berth system [4]), numerous studies have been published on BAP aiming to optimize the operation efficiencies by taking into account the various constraints affecting BAP [2], such as tidal influence [5], vessel service priority [6], time-varying water depth [7], and channel flow control [8]. As to our knowledge, no study on BAPs with the DLSM mode is observed in the literature; thus, this study aims to fill the research gaps in this field by taking into account the DLSM mode when constructing berth schedules. Furthermore, this study is specifically dedicated to continuous BAP, which copes with the real situation of Yangshan deep-water port that we have investigated and can also be applied to many other huge container hubs.

With regard to the methodologies applied to BAPs, it can be observed in the literature that the BAPs are normally formulated as Mixed Integer Programming (MIP) models, which can be solved with commercial programming solvers for small-size problems [2–11]. Since BAPs are NP-hard, lots of researchers have also taken much effort on developing near-optimal solutions with efficient heuristics and meta-heuristics, such as genetic algorithm and its variants [12–15], subgradient optimization method [15], simulated annealing [9], evolutionary algorithms [16, 17], particle swarm optimization [10], and island-based meta-heuristic algorithm [18].

Particle swarm optimization (PSO) is an evolutionary computation technique firstly developed by Eberhart and Kennedy [19]. Since the particles used in PSO algorithm are only updated by internal velocity and fewer parameters should be tuned, the principles of PSO can be easily understood and widely adapted to various specific applications. According to the literatures, PSO algorithms have been proven quite efficient in dealing with berth allocation [20] yard allocation [21], quay crane scheduling [10, 22]; and many other planning and scheduling problems [23, 24]; in consequence, a PSO algorithm is also proposed in this study to develop efficient berth schedules by taking into account DLSM.

3. Problem Description

3.1. Assumptions. In general, the berth operators are in charge of arranging each vessel arriving at the port to a suitable berth position according to the availability of the wharf resources and respecting the preplanned arrival and departure of the vessel. According to the practices observed at the targeted container terminal, a set of assumptions are defined as follows:

(1) Each vessel arrives at the port on the preplanned arrival time

(2) If a vessel cannot leave the port before the preplanned departure time, the port has to incur a penalty

(3) The coordinate corresponding to the leftmost end of the vessel is used to represent its berthing position, using the leftmost boundary of the wharf as the coordinate reference point

(4) Each vessel has a predefined minimum-cost berthing position, which is determined according to the goods that will be loaded/unloaded at the port, and if a vessel cannot be berthed at its ideal berthing position, additional operation cost will occur

(5) DLSM mode is applied, i.e., at most, two vessels can be simultaneously moored at the same berth position

(6) When two vessels are moored in double-line, the length of the inner-side ship cannot be longer than the vessel moored outside

(7) When two vessels are moored in double-line, the inner-side one must be berthed earlier and leave later than the outside one

3.2. Notations. For a better understanding, two types of notations are applied in this study: (1) Latin letters are used to denote parameters; (2) Greek letters are applied to represent decision variables.

3.2.1. Parameters

V: set of the vessels waiting to be allocated at the berth;

L: length of the wharf at the port;

T: planning horizon of this study;

M: sufficiently large positive number;

i, j, k: indices of vessels $i, j, k \in V$;

c_{1i}: unit distance cost for transporting containers from/to vessel i, $i \in V$;

c_{2i}: penalty cost per unit of time caused by the late departure of vessel i from the port, $i \in V$;

p_i: predefined minimum-cost berthing position of vessel i, $i \in V$.

e_i^{arr}: preplanned arrival time of vessel i;

h_i: total handling time required by the port to finish the necessary unloading/unloading operations of vessel i, $i \in V$;

e_i^{dep} : preplanned departure time of vessel i, $i \in V$;

l_i: length of vessel i, $i \in V$. In this study, the necessary gap that must be reserved to guarantee the safety is also integrated in this value for each vessel;

3.2.2. Decision variables

μ_i: integer variable representing the actual berthing position of vessel i, $i \in V$;

θ_i: Integer variable representing the actual berthing time of vessel i, i.e., the moment that this vessel is moored at its berthing position where it can be operated by the cranes, $i \in V$;

$\sigma_{i,j}$: binary variable representing the relative berthing position of two adjacent vessels, which equals 1 if vessel i is moored on the left to vessel j $(i, j \in V)$ and 0 otherwise;

$\pi_{i,j}$: binary variable indicating the sequential relationship between the berthing time of two vessels, which equals 1 if vessel i is berthed before vessel j $(i, j \in V)$ and 0 otherwise;

$\varepsilon_{i,j}$: binary variable representing the relative line position of two vessels that are moored in double-line, which equals to 1 if vessel i is berthed on the inner-side to vessel j $(i, j \in V)$ and 0 otherwise.

3.3. Timeline Corresponding to the Berth Scheduling of a Vessel. As shown in Figure 1, the timeline of the berth scheduling of vessel i starts from its arrival time at the port, denoted as e_i^{arr}, and ends up at the moment when all the necessary unloading/loading operations are completed, and the vessel is ready to leave the port for the next destination.

It is worth noting that (1) a gap between the preplanned arrival time e_i^{arr} and actual berthing time θ_i may be observed in the condition that no berth location is available on the arrival of the vessel which has to be waiting at the anchorage until in-wharf permission is delivered; (2) although setup operations (such as berthing, mooring ropes, and removing twist locks) are necessary before and after the cranes operating the vessels, the setup time of vessel i is integrated into its overall operation time hi in this study to simplify the expression because it is observed in the targeted port that the setup time is generally not vessel-dependent.

3.4. Berth Scheduling with DLSM Mode. When the DLSM mode is applied at the port and two vessels will be scheduled to be moored in double-line at the same berthing position, it is necessary to determine the relative berthing line positions between these two vessels. Through the interview with the berth operators at the targeted port, a general rule is applied to ensure the berthing safety: Vessel i can be moored in the inner-side line to vessel $j (i, j \in V)$ as long as the following conditions can be satisfied: (1) the length of vessel i is larger than that of vessel j; (2) vessel i arrives no later than vessel j at the port; (3) the actual departure time of vessel i is not earlier than that of vessel j; (4) the berthing position of the inner-side vessel should not be larger that of the outside moored ship while the coordinate of the rightmost end of the former along the wharf must be at least as large as that of the latter.

Figure 2 is the three-dimensional schematic diagram of a DLSM example with three vessels, where vessels i and j are moored in double-line.

For a better understanding of this example, the corresponding side-wharf section and time-wharf one are detailed in Figures 3 and 4, respectively. It can be observed that vessel i is placed on the inner-side line to vessel j since the following conditions are satisfied (1) $l_i \geq l_j$ (as shown in Figure 3); (2) $\theta_i \leq \theta_j$ (as shown in Figure 4); (3) $\theta_i + h_i \geq \theta_j + h_j$ (as shown in Figure 4); (4) $p_i \leq p_j$, $p_i + l_i \geq p_j + l_j$ (as shown in Figure 4); in a berth schedule, a vessel can either be placed at its minimum-cost position (e.g., vessels j and k) or nonoptimal berthing position (e.g., vessel i).

As for vessel k, since it is moored on the right to the pair of double-lined ships, the coordinates of its berthing position must be larger than those of the rightmost of the inner-side moored ship to avoid overlaps, i.e., $\mu_k \geq \mu_i + l_i$.

4. Mathematical Formulation

4.1. Construction of the Objective Function. As mentioned in Section 1, the objective function of the targeted BAP problem consists of two parts:

(1) Minimization of the additional operation cost for vessels not located at its minimum-cost berthing position

Considering that the ports normally predefine the minimum-cost berthing position for each vessel according the cargos that will be unloaded/loaded to maximize the operation efficiency, it is obvious that the larger is the deviation between the berthing

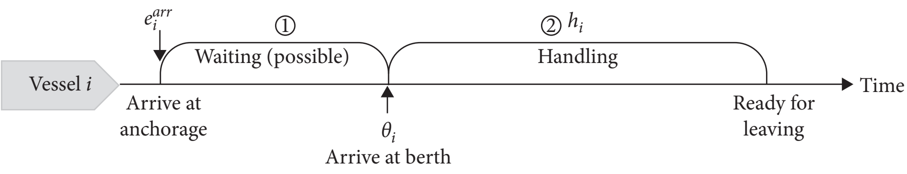

Figure 1: Timeline corresponding to the berth scheduling of vessel i.

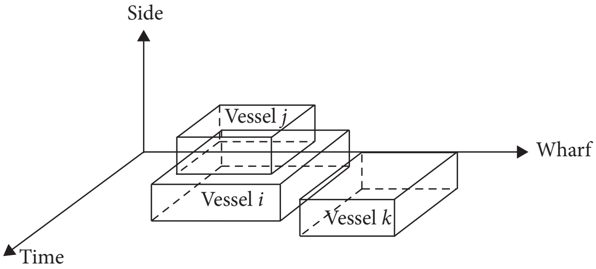

Figure 2: Time-side-wharf schematic diagram of a DLSM example.

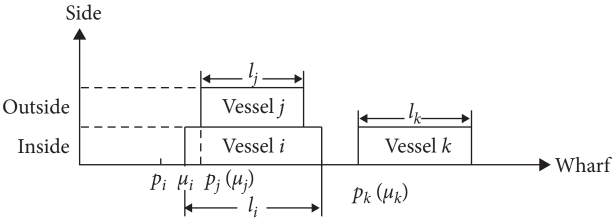

Figure 3: Side-wharf section of the DLSM example.

position and the predefined minimum-cost position of a vessel, the more is operation cost at the wharf; in this study, this part of the objective function is formulated as $\sum_{i\in V} c_{1i}|\mu_i - p_i|$.

(2) Minimization of the penalty cost for vessels not leaving before their preplanned departure time

Let $e_i^{de\,p}$ denote the preplanned departure time of vessel i. As is known, when the actual departure time of vessel i is later than $e_i^{de\,p}$, an additional cost will be incurred due to the influence of such delay on the rest of the voyage, and if the delay in the ship's departure is caused by the inefficient operation of the port, the port would have to pay certain fine for such delay. Therefore, this study aims also at minimizing the total penalties related to the delays of the vessels within the planning period, and this part of the objective function can be formulated as $\sum_{i\in V} c_{2i}(\theta_i + h_i - e_i^{de\,p})^+$, where $x^+ = \max\{0, x\}$.

To sum up, the objective function can be formulated as follows:

$$【BAP】\ \min \sum_{i\in V}\left\{c_{1i}\left|\mu_i - p_i\right| + c_{2i}\left(\theta_i + h_i - e_i^{de\,p}\right)^+\right\}. \quad (1)$$

4.2. Mixed-Integer Programming Model. Considering that formula (1) is nonlinear, it should be linearized to meet the requirements of linear programming solver that will be applied in this study to optimally solve small-size instances. Let $\alpha_i^+ = \mu_i - p_i$ when $\mu_i - p_i \geq 0$, $\alpha_i^- = p_i - \mu_i$ when $\mu_i - p_i < 0$, and β_i^+ and β_i^- denote the non-negative and negative value of $(\theta_i + h_i - e_i^{de\,p})$, respectively. Furthermore, let $\beta_i^+ = 0$ when $\theta_i + h_i - e_i^{\text{dep}} < 0$ to ensure that the smaller is β_i^+, the better is the solution. Formula (1) can be linearized to formulas (2), (3), and (4) and then the mixed-integer programming model of the targeted problem can be constructed as follows:

$$【BAP】\ \min \sum_{i\in V}\{c_{1i}(\alpha_i^+ + \alpha_i^-) + c_{2i}\beta_i^+\}, \quad (2)$$

subject to

$$\mu_i - p_i = \alpha_i^+ - \alpha_i^-, \quad \forall i \in V, \quad (3)$$

$$\theta_i + h_i - e_i^{de\,p} = \beta_i^+ - \beta_i^-, \quad \forall i \in V, \quad (4)$$

$$\sum_{j\in V}\varepsilon_{i,j} \leq 1, \quad \forall i \in V, \quad (5)$$

$$\mu_i + l_i \leq L, \quad \forall i \in V, \quad (6)$$

$$\mu_i + l_i \leq \mu_j + M\left(1 - \sigma_{i,j}\right), \quad \forall i, j \in V, i \neq j, \quad (7)$$

$$\theta_i + h_i \leq \theta_j + M\left(1 - \pi_{i,j}\right), \quad \forall i, j \in V, i \neq j, \quad (8)$$

$$\theta_i \leq \theta_j + M\left(1 - \varepsilon_{i,j}\right), \quad \forall i, j \in V, i \neq j, \quad (9)$$

$$\theta_j + h_j \leq \theta_i + h_i + M\left(1 - \varepsilon_{i,j}\right), \quad \forall i, j \in V, i \neq j, \quad (10)$$

$$\mu_i \leq \mu_j + M\left(1 - \varepsilon_{i,j}\right), \quad \forall i, j \in V, i \neq j, \quad (11)$$

$$\mu_j + l_j \leq \mu_i + l_i + M\left(1 - \varepsilon_{i,j}\right), \quad \forall i, j \in V, i \neq j, \quad (12)$$

$$\sigma_{i,j} + \sigma_{j,i} + \pi_{i,j} + \pi_{j,i} + \varepsilon_{i,j} + \varepsilon_{j,i} \geq 1, \quad \forall i, j \in V, i \neq j, \quad (13)$$

$$\sigma_{i,j} + \varepsilon_{i,j} \leq 1, \quad \forall i, j \in V, i \neq j, \quad (14)$$

$$\pi_{i,j} + \varepsilon_{i,j} \leq 1, \quad \forall i, j \in V, i \neq j, \quad (15)$$

$$\theta_i \geq e_i^{arr}, \quad \forall i \in V, \quad (16)$$

$$\mu_i, \theta_i, \alpha_i^+, \alpha_i^-, \beta_i^+, \beta_i^- \geq 0, \quad \forall i \in V, \quad (17)$$

$$\sigma_{i,j}, \pi_{i,j}, \varepsilon_{i,j}\varepsilon\{0, 1\}, \quad \forall i, j \in V, i \neq j. \quad (18)$$

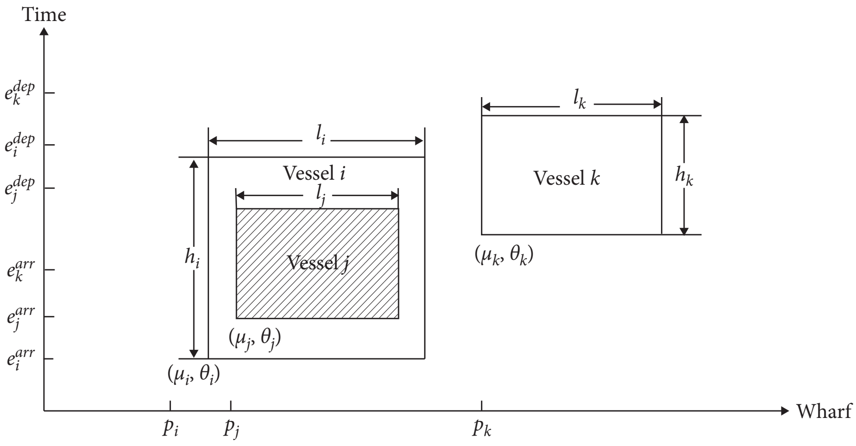

Figure 4: Time-wharf section of the DLSM example.

Objective function (2) and constraints (3) and (4) indicate that the objective of this study is to minimize both total additional operation cost corresponding to vessels not berthing at their minimum-cost berthing position vessels and the penalty cost incurred when vessels cannot leave before their preplanned departure time.

Constraint (5) ensures that no more than two vessels can be berthed at the same position simultaneously. Constraint (6) indicates that the rightmost end of each vessel must be limited by the length of the wharf. Constraint (7) ensures the position relationship of two adjacent vessels along the wharf. Constraint (8) indicates the sequential relationship between the berthing time of two vessels that will be berthed at the same position but not in double-line. Constraints (9)–(12) indicate the conditions to be respected when two vessels are berthed in double-line at the berth as detailed in Section 3.4. Constraints (13) ensure that at least one relationship between two vessels waiting to be berthed within the planning period, as shown in Figure 4, holds.

Constraints (14) and (15) ensure the SLSM mode and DLSM mode cannot be applied to the same pair of vessels simultaneously, i.e., if vessel i is berthed in double-line with vessel j, it can neither be berthed to the left of vessel j nor be berthed before the arrival or after the departure of vessel j. Constraint (16) ensures that vessels can only be berthed after their arrival. Constraints (17) and(18) define the range of decision variables.

5. PSO Algorithm for BAP with DLSM Mode

5.1. Introduction to PSO. In PSO algorithms, the particle swarm concept originated as a simulation of a simplified social system by introducing a number of simple entities—the particles—in the search space, where each particle represents a solution approach corresponding to a given position and velocity, which can be used to evaluate the objective function at its current location.

The movement of each particle is guided by their position, according to their own best position and a swarm's best position, which represents the quality of searching, and the velocity decides the direction in which the particle would move in the next generation. These particles search for optimal solutions through updating generations. Formulas (19) and (20) represent how velocity and position update in the classical PSO algorithm, respectively:

$$v_{id}^{k} = v_{id}^{k-1} + c_1 r_1 \left(p\text{best}_{id} - x_{id}^{k-1} \right) + c_2 r_2 \left(g\text{best}_{d} - x_{id}^{k-1} \right), \quad (19)$$

$$x_{id}^{k} = x_{id}^{k-1} + v_{id}^{k-1}, \quad (20)$$

where v_{id}^{k} and v_{id}^{k-1} represent the current and previous flight velocity of particle i on dimension d in iteration k, respectively, x_{id}^{k} and x_{id}^{k-1} represent the current and previous position of particle i on dimension d in iteration k, ω^{k-1} is the inertial weight coefficient, which can adjusts the search range of solution space, c_1 and c_2 are acceleration weights, which adjust the learning maximum step length, r_1 and r_2 are two random functions with a value range of [0, 1], whose function is to increase the randomness of the search. $p\text{best}_{id}$ denotes the best position of particle i on dimension d up to iteration k, while $g\text{best}_{d}$ denotes the best position of the whole swarm on dimension d until iteration k.

Considering that the classical PSO algorithm mentioned above may lead the particles to grow unlimitedly, which influences the particles' convergence to the optimal solution, Shi and Eberhart [25] improved the updating mechanism, by introducing an inertia weight coefficient, which can be dynamically adjusted to balance the quality of solution and convergence velocity of the algorithm, as shown in the following formula:

$$\begin{aligned} v_{id}^{k} = {} & \omega^{k-1} v_{id}^{k-1} + c_1 r_1 \left(p\text{best}_{id} - x_{id}^{k-1} \right) \\ & + c_2 r_2 \left(g\text{best}_{d} - x_{id}^{k-1} \right), \end{aligned} \quad (21)$$

where $\omega^{k-1} = [(\gamma_{\max} - \gamma_{k-1})/\gamma_{\max}] * (\omega_{\max} - \omega_{\min}) + \omega_{\min}$, which is the inertia weight coefficient, $\omega_{\max}$ and $\omega_{\min}$ denote the maximum and minimum values of the inertia weight coefficient, respectively, and $\gamma_{\max}$ represents the maximum number of iterations.

The PSO algorithm proposed in this study is based on the updating mechanism proposed by Shi and Eberhart (1998).

5.2. Encoding. Assuming that n vessels are waiting to be scheduled within the planning period, n random numbers, denoted as τ_i, $i \in \{1, \ldots, n\}$, are randomly generated in the range of 0 and 1.0, where each random number corresponds to the vessel with the same index.

Sort those generated random numbers in descending order and then allocate the corresponding vessels to berth positions one after another, i.e., the greater the random number τ_i is, the earlier vessel i is allocated to a berth position. Ties are broken by selecting the vessel with smallest index.

For a better understanding, here illustrated in Figure 5 is the encoding process with an example of 5 vessels, where a solution with the corresponding berthing order of the vessels as 5-3-1-4-2 is obtained.

Step 1: Generate a random number for each vessel.

Vessel	1	2	3	4	5
τ_i	0.2	0.1	0.4	0.2	0.8

⇩

Step 2: sort τ_i in descending order.

Vessel	5	3	1	4	2
τ_i	0.8	0.4	0.2	0.2	0.1

Figure 5: Schema of the encoding process.

5.3. Decoding. The decoding process, which is applied in this study to construct the berth schedule corresponding to a given solution obtained by the proposed PSO algorithm, consists of three steps as follows:

(i) Step 1: initialization of the berthing schedule

The initial berth schedule can be generated by arranging each vessel one after another in the order defined by the solution to its minimum-cost berthing position. It is worth mentioning that although placing vessels to their pre-defined minimum-cost berthing positions can avoid additional operation costs, it is hardly possible for berth operators to arrange all the vessels to their minimum-cost berthing positions without overlapping any of them at a busy port. In consequence, there is a good chance that the berth schedule obtained at this step is infeasible due to the overlaps, and therefore, actions have to be taken to detect and resolve possible overlaps.

(ii) Step 2: detection of overlaps

Considering that the overlap between two vessels takes place if and only if both berthing periods and spaces of these two vessels are partly overlapped; the overlap between two vessels j and k $(j, k \in V)$ can be detected by verifying constraints (22)–(25); in Figure 6, an example of three vessels with overlap detected between two vessels j and k is shown:

$$\mu_k < \mu_j + l_j, \quad k, j \in V, \tag{22}$$

$$\theta_k < \theta_j + h_j, \quad k, j \in V, \tag{23}$$

$$\mu_j < \mu_k + l_k, \quad k, j \in V, \tag{24}$$

$$\theta_j < \theta_k + h_k, \quad k, j \in V. \tag{25}$$

(iii) Step 3: overlaps resolving

Once overlaps are detected, the current berth schedule is not yet feasible, and thus actions must be taken to remove those overlaps. The procedure resolving overlaps between two vessels j and k $(j, k \in V)$ is as follows:

Step 3.1: removing overlap detected between two vessels

In this study, the overlap detected between two vessels is eliminated by fixing one vessel and moving the other one towards all possible directions until no overlap is observed between them. Here shown in Figure 7 is an example with two overlapped vessels, which are represented with solid line rectangles. Let vessel j be fixed and vessel k can be moved towards four possible directions to eliminate the overlap: (i) left (in condition that the leftmost end of vessel k does not exceed the left end of the wharf), (ii) right (in condition that the rightmost end of vessel k does not exceed the right end of the wharf), (iii) up (to delay its berthing time), and (iv) outside (in condition that the DLSM constraints are satisfied). The possible positions of vessel k after performing these movements are mentioned with dashed line rectangles, and the rectangle corresponding to the outside movement is shaded on this time-wharf section.

Upon further analysis of the four movements mentioned above, it can be observed that only movement (iii) can result in a feasible solution because (1) movement (i) is not available because there is not enough space on the left (dashed rectangle exceeds the left boundary of the wharf); (2) movement (ii) introduces an overlap between vessel k and vessel i; (3) movement (iv) is not available as well because the constraints related to the DLSM mode, as described in Section 4, cannot be satisfied. Nevertheless, the feasibility of the solution obtained by (ii) can be improved by taking into account the relationship between the vessel being moved, i.e., vessel k, and the nearby vessels that may be overlapped by the newly placed vessel k, e.g., vessel i in the example shown in Figure 7.

Step 3.2: improving the feasibility of berthing schedule by taking into account the nearby vessels having overlaps with certain moved vessel

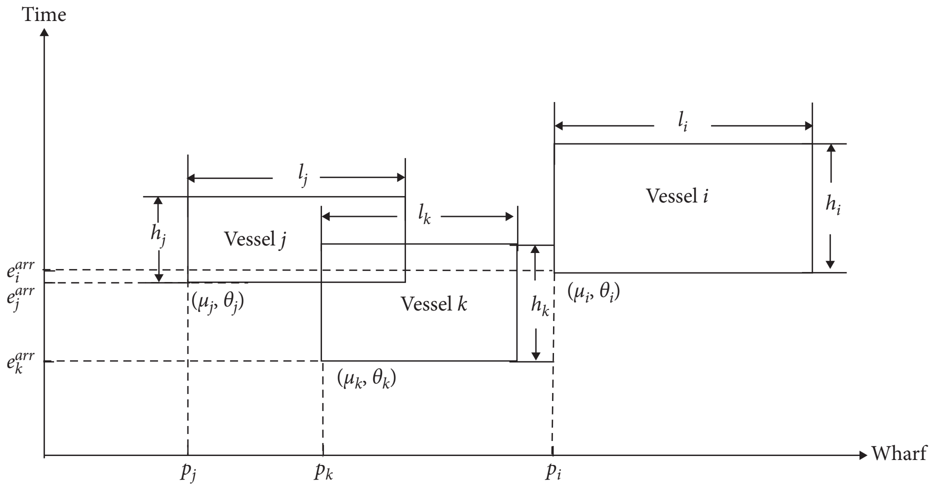

Figure 6: An example of three vessels obtained at step 1 with overlap detected.

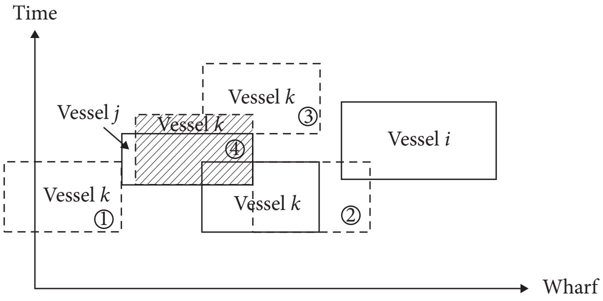

Figure 7: Illustration of possible movements made to remove the overlap between two vessels in time-wharf section.

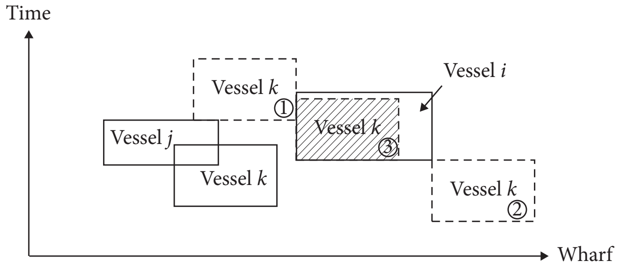

Figure 8: Possible movements of vessel k around the nearby vessel i.

Since it is possible to introduce new overlaps between the vessel being moved and some of the nearby ships, the relationship of all vessels that may have overlaps with the newly placed vessel must be considered to avoid introducing new overlaps.

For a better understanding, let us continue the illustration with the example mentioned in step 3.1. Since moving vessel k towards right may introduce an overlap between the vessels k and i, the movements of vessel k around vessel i are also considered to generate possible feasible berthing schedules. As shown in Figure 8, three new berthing schedules can obtained by finding the optimal position of placing vessel k adjacent to vessel i in condition that it does not overlap with any other vessels. It is worth noting that the berthing schedule corresponding to the optimal position above vessel i is not shown in Figure 8 because that berthing schedule can be dominated by at least the one with vessel k on lower left, i.e., the berthing schedule corresponding to movement (i) shown in Figure 8.

Step 3.3: accepting the best feasible berthing schedule

Compare all of the possible feasible berthing schedules generated by the adjustments described in steps 3.1 and 3.2 and accept the best one, i.e., the feasible berthing schedule with the smallest objective value as the one that corresponds to the given solution obtained by the proposed PSO algorithm.

5.4. General Procedure of the Proposed PSO Algorithm.

The general procedure of the proposed PSO algorithm is as follows.

(i) Step 1: set up the parameters of the PSO algorithm, such as the number of particles and the value of inertia weight coefficient.

(ii) Step 2: initialize the position and velocity in allowable ranges for each particle and set iteration $k = 1$.

(iii) Step 3: calculate the fitness value, which is equal to the objective value of the proposed model, for each particle.

(iv) Step 4: set the local-best value and global-best value for each particle, where the former equals the particle's current position and the latter the position of the best particle.

(v) Step 5: update the velocity and the position for each particle.

(vi) Step 6: update the fitness value for each particle.

(vii) Step 7: compare the current fitness value of each particle with the local-best one. If the current fitness value of a particle is better, update the local-best position of this particle; otherwise, it remains unchanged.

(viii) Step 8: find out the particle with the best fitness function from the current swam. If the current best fitness value is better than that of the recorded global-best one, replace the global-best position with the position of the current best particle; otherwise, the global-best one remains unchanged.

(xi) Step 9: if the number of iteration k attains the predefined threshold, the PSO algorithm terminates and reports the recorded global-best particle as the final solution; otherwise, set $k = k + 1$ and return to step 3.

6. Experimental Results

6.1. Experimental Settings. In this study, instances of different scales are randomly generated with the method introduced by Park and Kim [15]. The length of wharf is set as 1200 meters. The planning horizon T is set as 120 time units, where the time unit is one hour.

The cost coefficients c_{1i} and c_{2i} are set as 2 and 10, respectively, as proposed by Meisel and Bierwirth [26]. In order to ensure that most of the vessels can leave the port before their preplanned departure time, the value of the preplanned departure time of a vessel is determined by adding 1.0 to 2.0 times of the corresponding operation time to its preplanned arrival time, i.e., $e_i^{dep} = e_i^{arr} + h_i * q \ (i \in V)$ and q is a decimal randomly generated between 1.0 and 2.0, as proposed by Park and Kim [15]. The generation of the other parameters is detailed in Table 1.

The numerical experiments are programmed in C# (VS2017) on a PC with 2.3 GHz Intel Core i5 CPU and 4 GB RAM, and CPLEX 12.5 is applied as the programming solver for small-size instances. Both programming solver and the proposed PSO algorithm are set to terminate within 3 hours (10,800 s).

6.2. Comparison between Different Mooring Modes. First of all, experiments are conducted to compare between two different mooring modes, i.e., DLSM mode and SLSM mode by considering both objective values and execution time for small-size instances, i.e., the instances with up to 25 vessels.

As shown in Table 2, it can be observed that the optimal solutions for both modes can be obtained by CPLEX solver within 5 seconds for the instances with no more than 15 vessels. As for SLSM mode, the execution time used to solve instances with the SLSM mode by the CPLEX solver (as shown in column "CPU1") increases dramatically when the number of vessels is beyond 20. When the DLSM mode is applied, solution for instances with up to 20 vessels can be obtained within 10 seconds and the instances with 25 vessels can still be obtained within 30 minutes (as shown in column "CPU2"). As shown in column "Diff_CPU," the different rate of the execution time (Diff_CPU1 = (CPU2 − CPU1)/CPU1 * 100%) varies from −79.16% to −98.87% for the instances with 20 and 25 vessels, and it is reasonable to conclude that the application DLSM mode can greatly improve the work efficiency of port operators.

Table 1: Parameters used in the experiments.

Parameter	Distribution type	Range
e_i^{arr}	Uniform distribution	$U(1, 96)$
h_i	Uniform distribution	$U(10, 24)$
p_i	Uniform distribution	$U(1, 1200)$
l_i	Uniform distribution	$U(150, 350)$

With regard to the objective values, it can be observed that the DLSM mode obviously dominates the SLSM mode because the objective values of solutions with the DLSM mode (shown in column "OBJ2") are at least as good as those with the SLSM mode (shown in column "OBJ1"). According to the difference rates shown in column "Diff_Obj1" (Diff_Obj1 = (OBJ2 − OBJ1)/OBJ1 * 100%), operation costs can be reduced in average of 20.35% and the maximum reduction rate reaches 37.14%.

To sum up, it can be concluded that DLSM mode can help the port operators in not only improving their work efficiency but also reducing overall operation costs.

It should also be mentioned that the optimal solutions cannot be obtained by CPLEX solver within 3 hours for the instances with more than 25 vessels for neither of these two modes. Therefore, we can conclude that CPLEX solver is only effective for solving small-scale problems regardless of whether DLSM is applied, and thus it is necessary to develop efficient heuristics to obtain good quality solution within reasonable execution time for large-scale instances so as to cope with the real requirements of the huge terminal containers such as Yangshan port.

6.3. Comparison between Different Methodologies. As mentioned before, the CPLEX solver is just capable of solving BAP models for small-scale instances with both SLSM and DLSM modes, though much more vessels must be scheduled during even 120 hours. Thus, in this study, a PSO algorithm has been proposed to obtain good quality solutions within reasonable execution time for large-scale instances.

As shown in Table 3, when comparing the solutions obtained by CPLEX solver and the proposed PSO algorithm for instances with DLSM modes, we can observe that both CPLEX solver and the proposed PSO algorithm can get the final solution very quickly for the instances within 20 vessels. As for instances with more vessels, the CPLEX solver becomes more and more inefficient and cannot obtain optimal solutions within three hours for instances with beyond 30 vessels, though PSO can still get final solution within several minutes.

With regard to objective values, the proposed PSO can obtain optimal solutions for the instances with 8 vessels and most of the instances with 10 vessels and even one instance with 20 vessels; near-optimal solutions can be obtained for the rest of the instances with 10 vessels and most of the cases with 15 vessels and even most of the cases with 25 vessels

Table 2: Comparison between DLSM and SLSM modes for small-scale instances.

Instances	SLSM		DLSM		Diff_Obj1 (%)	Diff_CPU1 (%)
	OBJ1	CPU1 (s)	OBJ2	CPU2 (s)		
8-1	300	0.2	252	0.2	−16.00	0.00
8-2	324	0.3	304	0.2	−6.17	−33.33
8-3	488	0.3	488	0.4	0.00	33.33
10-1	430	0.3	430	0.5	0.00	66.67
10-2	656	0.4	656	0.3	0.00	−25.00
10-3	754	0.6	474	0.5	−37.14	−16.67
15-1	2642	4.4	2020	2.3	−23.54	−47.73
15-2	804	2.2	544	2.3	−32.34	4.55
15-3	940	0.5	940	1.1	0.00	120.00
20-1	2816	733.2	1842	8.3	−34.59	−98.87
20-2	3574	205.2	2604	7.7	−27.14	−96.25
20-3	1932	116.4	1232	3.6	−36.23	−96.91
25-1	4532	3630	3384	668.7	−25.33	−81.58
25-2	6886	7245.2	4616	1510.1	−32.97	−79.16
25-3	4440	7280.1	2936	1421.0	−33.87	−80.48
				Average	−20.35	−28.76

Table 3: Comparison between the performance of CPLEX and PSO algorithm for solving problems with the DLSM mode.

Instances	CPLEX		PSO		Diff_Obj2 (%)
	OBJ2	CPU2 (s)	OBJ3	CPU3 (s)	
8-1	252	0.2	252	8.2	0.00
8-2	304	0.2	304	4.4	0.00
8-3	488	0.4	488	9.4	0.00
10-1	430	0.5	430	10.1	0.00
10-2	656	0.3	656	8.0	0.00
10-3	474	0.5	484	16.2	2.11
15-1	2020	2.3	2052	40.7	1.58
15-2	544	2.3	574	41.3	5.51
15-3	940	1.1	1066	23.6	13.40
20-1	1842	8.3	2102	73.7	14.12
20-2	2604	7.7	2974	73.9	14.21
20-3	1232	3.6	1232	73.6	0.00
25-1	3384	668.7	3904	103.3	15.37
25-2	4616	1510.1	4854	122.8	5.16
25-3	2936	1421.0	3022	111.3	2.93
30-1	Cannot obtain the optimal solution	>3 h	7566	326.3	—
30-2		>3 h	6820	251.1	—
30-3		>3 h	8248	211.4	—
35-1			7898	272.2	—
35-2			12006	273.1	—
35-3			12760	279.5	—
40-1			15800	354.1	—
40-2			15640	565.2	—
40-3			20558	474.2	—
45-1			26150	685.2	—
45-2			35374	744.6	—
45-3			31726	644.2	—

with quite small difference rate, which can be illustrated in column "Diff_Obj2" (Diff_Obj2 = (OBJ3 − OBJ2)/OBJ2 ∗ 100%). It hints that the proposed PSO algorithm is also possible to get solutions of good quality for large-scale instances though further studies should be made to test the condition of such performance.

Table 4: Comparison between optimal solutions with SLSM mode and solutions obtained by PSO with the DLSM mode.

Instances	CPLEX-SLSM		PSO-DLSM		Diff_Obj3 (%)
	OBJ1	CPU1 (s)	OBJ3	CPU3 (s)	
8-1	300	0.2	252	8.2	−16.00
8-2	324	0.3	304	4.4	−6.17
8-3	488	0.3	488	9.4	0.00
10-1	430	0.3	430	10.1	0.00
10-2	656	0.4	656	8.0	0.00
10-3	754	0.6	484	16.2	−35.81
15-1	2642	4.4	2052	40.7	−22.33
15-2	804	2.2	574	41.3	−28.61
15-3	940	0.5	1066	23.6	13.40
20-1	2816	733.2	2102	73.7	−25.36
20-2	3574	205.2	2974	73.9	−16.79
20-3	1932	116.4	1232	73.6	−36.23
25-1	4532	3630	3904	103.3	−13.86
25-2	6886	7245.2	4854	122.8	−29.51
25-3	4440	7280.1	3022	111.3	−31.94
Average					**−16.61**

Since it is observed in Table 3 that the gap between solutions obtained by the CPLEX solver and the PSO algorithm with the DLSM mode is relatively significant for some of the instances, a further comparison is made between the results obtained by PSO with DLSM mode and the optimal solutions obtained by the CPLEX solver with the SLSM mode.

As shown in Table 4, solutions obtained by the PSO algorithm with DLSM mode are better than the optimal solutions obtained by the CPLEX solver with the SLSM mode, and the former can save up to 35.81% of the cost (Diff_Obj3 = (OBJ3 − OBJ1)/OBJ1 ∗ 100%) among all the instances tested in this study.

Considering that hundreds of vessels should be operated every day at huge container terminals, the proposed PSO will be much more practical than CPLEX for supporting the decision-making of the port operators to not only improve their working efficiency but also reduce operation costs related to berth scheduling operations.

7. Conclusions and Perspectives

The study aims at minimizing the total operation cost of the continuous berth scheduling problem by taking into account the Double-Line Shipping Mooring (DLSM) mode, where both the additional operation cost for vessels not moored at their minimum-cost berthing position and penalty cost related to vessels not being able to leave before its preplanned departure time are considered.

The problem is firstly formulated as a mixed integer programming model and solved by the CPLEX solver for small-scale instances. As for larger size instances that cannot be optimally solved by CPLEX solver, a PSO algorithm is proposed to obtain good quality solutions within reasonable execution time.

Numerical experiments are conducted to compare not only the efficiency between the traditional Single-Line Shipping Mooring (SLSM) mode and the innovative DLSM mode but also the performances between CPLEX solver and the proposed PSO algorithm. It can be concluded with the experimental results that (1) DLSM mode outperforms the SLSM mode in reducing not only total operation cost but also execution time. (2) The proposed PSO algorithm can generate optimal or near-optimal solution for small-scale instances. (3) The proposed PSO algorithm is much more efficient than the CPLEX solver for large-scale instances, which copes with the requirements of berthing management in Yangshan Deep-Water Port, one of the busiest container terminals in the world.

To sum up, as the first research dedicated to BAP with DLSM mode, this study can help not only in validating the advantages of DLSM mode but also offering an efficient decision support tool to berth operators in busy ports to improve their working efficiency.

Motivated by the results obtained in this study, it is interesting to keep improving the efficiency of the proposed algorithm and to apply such method in the targeted port.

Data Availability

All the experimental data can be generated with the rules described in the paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Q. Meng, S. Wang, H. Andersson, and K. Thun, "Containership routing and scheduling in liner shipping: overview and future research directions," *Transportation Science*, vol. 48, no. 2, pp. 265–280, 2014.

[2] D. Kizilay and D. T. Eliiyi, "A comprehensive review of quay crane scheduling, yard operations and integrations thereof in container terminals," *Flexible Services and Manufacturing Journal*, 2020.

[3] A. Imai, K. I. Nagaiwa, and C. W. Tat, "Efficient planning of berth allocation for container terminals in Asia," *Journal of Advanced Transportation*, vol. 31, no. 1, pp. 75–94, 1997.

[4] A. Imai, E. Nishimura, and S. Papadimitriou, "The dynamic berth allocation problem for a container port," *Transportation Research Part B: Methodological*, vol. 35, no. 4, pp. 401–417, 2001.

[5] V. H. Barros, T. S. Costa, A. C. M. Oliveira, and L. A. N. Lorena, "Model and heuristic for berth allocation in tidal bulk ports with stock level constraints," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 606–613, 2011.

[6] L. Dai and L. Tang, "Berth allocation with service priority for container terminal of hub port," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Logs Engineering & Management, Dalian, China, October 2008.

[7] T. Qin, Y. Du, and M. Sha, "Evaluating the solution performance of IP and CP for berth allocation with time-varying water depth," *Transportation Research Part E: Logistics and Transportation Review*, vol. 87, pp. 167–185, 2016.

[8] L. Zhen, Z. Liang, D. Zhuge, L. H. Lee, and E. P. Chew, "Daily berth planning in a tidal port with channel flow control," *Transportation Research Part B: Methodological*, vol. 106, pp. 193–217, 2017.

[9] K. H. Kim and K. C. Moon, "Berth scheduling by simulated annealing," *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 541–560, 2003.

[10] B. C. Jos, M. Harimanikandan, C. Rajendran, and H. Ziegler, "Minimum cost berth allocation problem in maritime logistics: new mixed integer programming models," *Indian Academy of Sciences/Sādhanā*, vol. 44, p. 149, 2019.

[11] L. Zhen, H. Hu, W. Wang, X. Shi, and C. Ma, "Cranes scheduling in frame bridges based automated container terminals," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 369–384, 2018.

[12] E. Lalla-Ruiz, J. L. González-Velarde, B. Melián-Batista, and J. M. Moreno-Vega, "Biased random key genetic algorithm for the tactical berth allocation problem," *Applied Soft Computing*, vol. 22, pp. 60–76, 2014.

[13] E. Nishimura, A. Imai, and S. Papadimitriou, "Berth allocation planning in the public berth system by genetic algorithms," *European Journal of Operational Research*, vol. 131, no. 2, pp. 282–292, 2001.

[14] S. R. Seyedalizadeh Ganji, A. Babazadeh, and N. Arabshahi, "Analysis of the continuous berth allocation problem in container ports using a genetic algorithm," *Journal of Marine Science and Technology*, vol. 15, no. 4, pp. 408–416, 2010.

[15] Y.-M. Park and K. H. Kim, "A scheduling method for berth and quay cranes," *OR Spectrum*, vol. 25, no. 1, pp. 1–23, 2003.

[16] M. A. Dulebenets, "Application of evolutionary computation for berth scheduling at marine container terminals: parameter tuning versus parameter control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 25–37, 2018.

[17] M. A. Dulebenets, "An adaptive island evolutionary algorithm for the berth scheduling problem," *Memetic Computing*, vol. 12, no. 1, pp. 51–72, 2020.

[18] M. Kavoosi, M. A. Dulebenets, O. Abioye et al., "Berth scheduling at marine container terminals: a universal island-based metaheuristic approach," *Maritime Business Review*, vol. 5, no. 1, pp. 30–66, 2020.

[19] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceeding of the 6th International Symposium on Micromachine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[20] C.-J. Ting, K.-C. Wu, and H. Chou, "Particle swarm optimization algorithm for the berth allocation problem," *Expert Systems with Application*, vol. 41, no. 4, pp. 1543–1550, 2014.

[21] L. Zhen, “Modeling of yard congestion and optimization of yard template in container ports,” *Transportation Research Part B: Methodological*, vol. 90, pp. 83–104, 2016.

[22] P. Guo, W. Cheng, and Y. Wang, “A modified generalized extremal optimization algorithm for the quay crane scheduling problem with interference constraints,” *Engineering Optimization*, vol. 46, pp. 1411–1429, 2014.

[23] H.-P. Hsu and C.-N. Wang, “Resources planning for container terminal in a maritime supply chain using multiple particle swarms optimization (MPSO),” *Mathematics*, vol. 8, no. 5, p. 764, 2020.

[24] M. Zhong, Y. Yang, Y. Zhou, and O. Postolache, “Adaptive autotuning mathematical approaches for integrated optimization of automated container terminal,” *Mathematical Problems in Engineering*, vol. 2019, Article ID 7641670, 14 pages, 2019.

[25] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE world congress on Computational Intelligence*, pp. 69–73, Anchorage, AK, USA, 1998.

[26] F. Meisel and C. Bierwirth, “Heuristics for the integration of crane productivity in the berth allocation problem,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 196–209, 2009.