Research Article

Particle Swarm Optimization Using Neighborhood-Based Mutation Operator and Intermediate Disturbance Strategy for Outbound Container Storage Location Assignment Problem

Yuyan He ¹, Aihu Wang ¹, Hailiang Su,² and Mengyao Wang¹

¹School of Business Administration, South China University of Technology, Guangzhou 510640, China

²School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China

Correspondence should be addressed to Aihu Wang; bmawang516@hotmail.com

Received 10 April 2019; Revised 18 June 2019; Accepted 27 June 2019; Published 9 July 2019

Academic Editor: Luciano Mescia

Copyright © 2019 Yuyan He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Outbound container storage location assignment problem (OCSLAP) could be defined as how a series of outbound containers should be stacked in the yard according to certain assignment rules so that the outbound process could be facilitated. Considering the NP-hard nature of OCSLAP, a novel particle swarm optimization (PSO) method is proposed. The contributions of this paper could be outlined as follows: First, a neighborhood-based mutation operator is introduced to enrich the diversity of the population to strengthen the exploitation ability of the proposed algorithm. Second, a mechanism to transform the infeasible solutions into feasible ones through the lowest stack principle is proposed. Then, in the case of trapping into the local solution in the search process, an intermediate disturbance strategy is implemented to quickly jump out of the local solution, thereby enhancing the global search capability. Finally, numerical experiments have been done and the results indicate that the proposed algorithm achieves a better performance in solving OCSLAP.

1. Introduction

Scientific storage location assignment (SLA) is a key factor to improve port efficiency. Economic globalization, transport containerization, container ship scale-up, and port automation in the past 4 decades have resulted in a significance restructuring of global container ports. Many major international container ports have been booming; e.g., the top three global container ports of Shanghai (China), Singapore (Singapore), and Shenzhen (China) handled 42.01 million TEUs, 33.6 million TEUs, and 25.74 million TEUs, respectively, in 2018, and enjoyed annual increase of 4.4%, 8.7%, and 2.1% accordingly. At the same time, many other container ports have been facing grand challenges; e.g., port of Kaohsiung handled 10.45 million TEUs with an annual increase of mere 1.7%, while port of Hamburg handled 9 million TEUs with an annual increase of -0.8% in 2018. The slow global economy recovery since the recess of 2008, the ever-increasing international trade protectionism, and disputes render fiercer container port competition. As a pivotal component of any container port, the storage has

been attracting more attention recently, since the storage operations, especially storage location assignment, not only streamline the quay operations, the horizontal transport operations, and the gate operations, but also affect directly the overall efficiency and profitability of the port. Therefore, storage location assignment mechanisms that cater to the need of extra-large container vessels and automated container ports are in great demand.

Outbound container storage location assignment (OCSLA) is the preparation work before shipment, which is closely related to the loading efficiency and waiting time of vessels in port. In general, outbound containers usually arrive at the storage yard within three to seven days before the ship arrives at the port. And the load planners will arrange the corresponding storage location according to quantity, size, weight, cargo type, and ship destinations of containers. However, due to the long time span of outbound containers entering into the storage yard and the uncertainty of arriving container quantity distribution, it is more difficult to allocate the detailed location of outbound containers. Unreasonable

location assignment will lead to an increase in the rate of container reshuffle and in waiting time for quay crane, thus reducing the loading efficiency of vessels and resulting in huge pressure on outbound operations. So, it is of practical significance to solve the problem of OCSLA in a scientific way.

Outbound container storage location assignment problem (OCSLAP) is a classic constraint combination problem that has been proven to be an NP-hard problem [1]. When solving such problems, the solution space will explode as the scale expands in the search area. Thereby, metaheuristic algorithms are applied to find the feasible solutions in a reasonable time. As a classical metaheuristic algorithm, PSO is similar to other evolutionary algorithms such as genetic algorithm (GA) [2, 3], ant colony optimization algorithm (ACO) [4], firefly algorithm (FA) [5], and artificial fish swarm algorithm (AFSA) [6]. The common characteristic of above-mentioned algorithms is to simulate a certain feature of a natural creature to search randomly through the solution space. Compared with other algorithms, PSO has fewer control parameters, better convergence, and better ability of searching global solution and is easy to implement, and hence it is suitable to solve the combinatorial optimization problems [7, 8].

To our knowledge, when PSO pursues the global optimal solution, it tends to ignore some local optimal solutions, which leads to fast convergence [9]. So, it needs to take measures to strengthen PSO's local search. In [10–12], local search strategies were introduced to find the optimal value in which the search direction of particles was controlled and aimed to balance the diversity of the population and the convergence speed of the particle swarm. Similarly a neighborhood-based mutation operator [13] is introduced to enhance the population diversity. However, when the neighborhood radius is too small, it is prone to be trapped in local solution. So, an intermediate disturbance strategy is adopted to jump out of the local optima and to strengthen the global exploration ability.

The main contributions of this article could be summarized as follows:

- (1) A new PSO algorithm is developed to solve OCSLAP, in which a neighborhood-based mutation operator and an intermediate disturbance strategy are introduced to balance the local exploitation and global exploration ability.

- (2) Infeasible solutions are transformed into feasible ones through a repairing strategy in which all the particles are repeatedly initialized or constantly adjusted according to the constraints until all the constraints are satisfied during initialization. At the same time, the lowest stack principle is introduced into the repairing strategy to balance the number of containers among different bays.

- (3) A series of small and medium scale OCSLAPs are solved to illustrate the effectiveness of the proposed algorithm. And results also prove that the proposed PSO algorithm performs better than GA and traditional PSO.

The rest of this paper is organized as follows. A systematic review of OCSLAP and particle swarm optimization methodology in the literature is provided in Section 2. Section 3 summarizes the assumptions, determines key decision variables, and then establishes a mathematical model.

Then, the new PSO algorithm is proposed in Section 4 and numerical experiments are addressed in Section 5. Finally, Section 6 presents the concluding remarks and potential research directions.

2. Literature Reviews

In recent years, OCSLAP has attracted considerable practical and academic attention due to a series of challenges such as the uncertainty of loading sequences, irregular distribution of outbound containers, temporary changes to the stowage plan, and the ever-increasing size of container ships.

In [14], considering the uncertainty of weight distribution for containers, a dynamic programming model was formulated to determine the specific location to minimize the number of reshuffles for loading operation. In [15], the authors improved the model proposed by Kim et al. [14], in which corresponding decision rules were derived based on the transformed dynamic programming model, aiming at minimizing the number of reshuffles. Based on the model proposed by Kim et al. [14] and Zhang et al. [15], Zhang et al. proposed two conservative allocation models in which rollout algorithm and a group-based stacking (GS) were used to solve models. Numerical experiments indicated that the improved model was superior to the original model of Kim et al. [14] and Zhang et al. [15]. In [16], a mixed integer programming model was established with the goal of minimizing the travel distance and the imbalance of the number of outbound containers in the first stage. In the second stage, a hybrid sequence stacking algorithm (HSSA) was applied to solve OCSLAP. From the results, it could be seen that HSSA was better than random stacking algorithm (RSA) and vertical stacking algorithm (VSA). In [17], the authors studied the detailed location of outbound containers at storage yard and computed the number of relocations movements assuming the worst case scenario and the distance travelled by internal trucks. Subsequently, a smart-relocation heuristic (SR) was proposed to solve this problem. In [18], the authors used the outer-inner cellular automaton method to solve the problem of choosing a certain bay and choosing proper slots for containers, in which these two problems were used as an integrated optimization process.

Different from the literature of Hu et al. [18] which goal was to minimize the number of reshuffles when solving the problem of choosing detailed location, this article only focuses on the problem of choosing proper slots, aiming at multiple influencing factors, including the number of reshuffles, the travel distance of internal truck, and the imbalance of the number of containers among bays. In addition, OCSLAP in this study is of combinatorial and NP-hard nature, so PSO, as a mature metaheuristic algorithm, is adopted to get satisfactory solution. There are two reasons why PSO is applied to solve OCSLAP. One is that PSO has been successfully applied to solve a series of combinatorial problems [19–21] and has been proven to achieve very efficient performance [22, 23]. The other is that, compared to other evolutionary algorithms such as GA and ACO, PSO has fewer operation parameters, easier adjustment, better ability of searching global solution, and faster convergence speed [24].

However, many literatures have also proved that PSO has a disadvantage of fast convergence [25–27]. Relevant literature studies have found that enhancing population diversity is an effective strategy to prevent premature of PSO algorithm. For instance, Gülcü et al. [28] proposed multi-swarm strategy and parallel strategy to enrich the diversity of the population. In [29], position mutation strategy was employed to retain the swarm diversity. Farnad et al. [30] applied a new hybrid method by a combination of three population based algorithms such as GA, PSO, and Symbiotic Organisms Search (SOS), in which mutualism phase was introduced to produce more warms. In [31], PSO could maintain a good diversity by performing crossover on the personal historical best position of each particle. Similarly, in [10, 32, 33], some strategies were also introduced to enhance population diversity. Consistent with the previous research, this study applies neighborhood-based mutation operator to balance the diversity of the population and convergence of PSO.

Executing neighborhood-based mutation operator can strengthen local exploitation abilities, but it also has the disadvantage of trapping into local solutions if the neighborhood radius is not chosen properly. Therefore, measures need to be taken to jump out of local solutions within a certain probability to achieve global optimality, which is good for maintaining the balance between local exploitation and global exploration abilities. Meanwhile, the existing literatures have also proposed various kinds of disturbance strategies. In [34], the genetic disturbance including simulated binary crossover and polynomial mutation was used to cross the corresponding particle in the external archive and generate new individuals, which would improve the swarm ability to escape from the local optima. Wang et al. [35] employed an adaptive mutation strategy to help trapped particles escape from local minima. Wu [36] used Cauchy operator to correct direction of particle velocity, improving the capacity of the global search of the algorithm. Similarly, Liang et al. [37] put forward a novel mutation disturbance to effectively avoid the phenomenon of falling into local optimal, in which all particles could search the solution space in larger range. Zhang et al. [25] applied the chaos optimization to make the improved PSO algorithm jumping out of the local optimal solution. In [38], the enhanced weighted quantum PSO (EWQPSO) was developed to perform the design of the supershaped lens antennas yielding optimal antenna performance. Since EWQPSO permitted all particles to have a quantum behavior and no velocities for particles are needed, the particles were guided to search the global optimal solution. Many improved approaches to escape from the local optima were also discussed in the literatures [39–42]. As far as the above literatures are concerned, two approaches could be taken to jump out of the local solution and hence to strengthen the global exploration ability: one is to interfere with the position of the particle swarm, and the other is to change the velocity of the particles. In this study, the first approach is adopted in which the position of the particle swarm is interfered by introducing an intermediate disturbance strategy to quickly escape from the local solution.

In conclusion, OCSLAP has been endowed with new meaning in the circumstance of container ship scale-up and container port automation, and research on PSO with neighbourhood-based mutation operator and intermediate disturbance strategy in OCSLAP domain is still in its infancy stage. Therefore, it is of both theoretical and practical significance to research on the proposed algorithm for OCSLAP.

3. Model Formulation

3.1. Problem Description. Storage operation of outbound containers refers to the detailed location assignment according to a series of rules before loading onto ships. Outbound containers enter from the gate or crossdocking ship and are transported directly by the special vehicle to container yard before ship's arrival. At the same time, the load planners will also arrange the corresponding location according to quantity, weight information, and ship destinations of outbound containers. In general, the stowage plan should not be disturbed at will as it will cause a large number of reshuffles, resulting in huge operation costs. So it is of great practical significance to solve OCSLAP by scientific approaches for the sake of port operation efficiency and competitiveness.

In a container yard, one block consists of many bays and one bay contains a series of stacks. Assume that a batch of outbound containers will be assigned to specific locations in two bays. As is shown in Figure 1(a), block A is composed of multiple bays from A1 to A8. Five outbound containers are B, C, D, E, and F, respectively, and their arrival sequence is 3, 1, 5, 2, and 4. Figure 1(b) shows the stacking status in bays A7 and A8 after outbound containers enter into the yard and the shaded squares represent the previously existing containers. From the above description, it could be concluded that finding the appropriate empty slots in each stack is the key to solve OCSLAP. For outbound containers location assignment, three important goals, which are the number of reshuffles, the distance between the internal truck and the berth, and the number of containers in different bays, should be considered so as to improve the loading efficiency and reduce waiting time of vessels.

In order to speed up the loading process, outbound containers are usually transported directly to the yard for location selection. First, the specific position in one bay determines the moving cost of the internal truck between in the yard side and quayside. Therefore, the first goal of this problem is to minimize the travel distance of the internal truck. Second, the order of picking up outbound containers is opposite to the stacking sequence, which greatly affects the loading efficiency of vessels. In the preparation for loading shipment, to obtain a better stacking strategy, it is necessary to reduce the number of reshuffles so that the containers can be loaded in the original loading order. So, the second goal of this problem is to minimize the number of reshuffles. Finally, when the detailed location of outbound containers is arranged, it is necessary to ensure that a series of equipment such as gantry cranes and trucks is protected from traffic congestion. To this end, the third goal is to maintain the balance of the number of containers among bays as uniformly as possible.

FIGURE 1: Detailed description of OCSLAP.

3.2. *Assumptions.* Mathematical models in this paper are established on the following assumptions:

(1) Outbound containers and inbound containers are not mixed up in the storage location assignment process. No matter the traditional Asian port layout where the queue line is parallel to the storage yard or the European port layout where the queue line is perpendicular to the storage yard which is adopted by most automated container ports, containers of different features are usually stored separately, e.g., empty and heavy containers, inbound and outbound containers, and twenty feet and forty feet containers. Therefore, the first assumption is valid even in automated container ports.

(2) Only one size of outbound containers (i.e., Forty Equivalent Unit, FEU) is considered. This assumption is based on the observation that containers of the same size are usually stored in the same bay and containers of different size in different bays. To facilitate the theoretical research, only FEU is considered in this paper.

(3) Container ship stowage plan is known in advance and does not change arbitrarily. In practice, the stowage plan is influenced by many factors like custom inspection quantity and speed and ship harboring process. Hence it is not fully determined until the ship is securely harbored in the port and the custom is completely cleared. However, more than 80-90% of the containers usually arrive in the port storage yard by then. Therefore, it does not hurt to assume that the stowage plan is known and determined in advance.

(4) The weight levels of outbound containers are known in advance before shipments and the arrival sequence of outbound containers is generated randomly which is in accordance with the practical statistical data analysis results of Guangdong-Hong Kong-Macao Greater Bay Area container ports.

(5) The incoming outbound containers satisfy the first-in-first-stacking discipline and the initial status of bays is empty.

(6) To ensure the stability of the vessel at the time of loading (the heavy container is loaded on the bottom layer as much as possible to lower the vessel's center of gravity), the container that is usually stacked on the storage yard should follow the principle that the heavier container is on the upper locations and the lighter container in the lower. If this principle is violated, reshuffles will occur, as manifested in Figure 2. (The larger numbers indicate weight levels and the subscript numbers arrival order of outbound containers.) The total number of reshuffles is 4.

	3 ₁₆				6 ₁₉
7 ₆	6 ₁₀	2 ₁₁	5 ₁₈		5 ₁₅
3 ₅	2 ₉	3 ₈	2 ₁₄	5 ₁₇	4 ₇
1 ₁	2 ₄	1 ₂	4 ₁₂	9 ₁₃	3 ₃
The number of reshuffles					
0	1	1	1	1	0

FIGURE 2: The number of reshuffles produced when outbound containers are assigned.

3.3. *Mathematical Modeling.* Notations used in this model are listed as follows.

Parameters

I Set of all containers ($i = 1, 2, \dots, I$)

B Set of all bays ($b = 1, 2, \dots, B$)

R Set of all stacks ($r = 1, 2, \dots, R$)

T Set of all tiers ($t = 1, 2, \dots, T$)

W_i The weight level of the i th container

d_{bi} The travel distance of internal truck when the i th container in the b th bay is loaded to the ship ($d_{bi} = 1, 2, \dots, B$)

$\text{Slot}(b, r, t)$ Slot of the b th bay, r th stack, and t th tier

V_b The volume of bay b

M A sufficiently large positive number (i.e., $M \gg 0$)

Decision Variables

x_{ibrt} Set to 1 if container i is stacked to $\text{Slot}(b, r, t)$, and 0 otherwise

$y_{br(t-1)}$ Set to 1 if containers a and b are, respectively, stacked in $\text{Slot}(b, r, t)$ and $\text{Slot}(b, r, t-1)$, and $W_a < W_b$, and 0 otherwise.

Z_b The workload of the b th bay

The OCSLAP model can be formulated as follows:

$$\text{Minimize } f_1 = \sum_{b=1}^B \left(d_{bi} \cdot \sum_{i=1}^I \sum_{r=1}^R \sum_{t=1}^T x_{ibrt} \right) \quad (1)$$

$$\text{Minimize } f_2 = \sum_{b=1}^B \sum_{r=1}^R \sum_{t=2}^T y_{br(t-1)} \quad (2)$$

$$\text{Minimize } f_3 = \max_{\{b\}} Z_b - \min_{\{b\}} Z_b \quad (3)$$

$$f = \min \left\{ \frac{f_1 - f_{1,\min}}{f_{1,\max} - f_{1,\min}} + \frac{f_2 - f_{2,\min}}{f_{2,\max} - f_{2,\min}} + \frac{f_3 - f_{3,\min}}{f_{3,\max} - f_{3,\min}} \right\} \quad (4)$$

$$\text{s.t. } \sum_{b=1}^B \sum_{r=1}^R \sum_{t=1}^T x_{ibrt} = 1, \quad i = 1, 2, \dots, I \quad (5)$$

$$\sum_{i=1}^I x_{ibr(t-1)} \geq \sum_{i=1}^I x_{ibrt}, \quad b = 1, 2, \dots, B, \quad r = 1, 2, \dots, R, \quad t = 2, 3, \dots, T \quad (6)$$

$$\sum_{i=1}^I x_{ibrt} \leq 1, \quad b = 1, 2, \dots, B, \quad r = 1, 2, \dots, R, \quad t = 1, 2, \dots, T \quad (7)$$

$$\sum_{i=1}^I \sum_{r=1}^R \sum_{t=1}^T x_{ibrt} \leq V_b, \quad b = 1, 2, \dots, B \quad (8)$$

$$Z_b = \sum_{i=1}^I \sum_{r=1}^R \sum_{t=1}^T x_{ibrt}, \quad b = 1, 2, \dots, B \quad (9)$$

$$\sum_{i=1}^I i \cdot x_{ibrt} + \left(1 - \sum_{i=1}^I x_{ibrt} \right) M \geq \sum_{i=1}^I i \cdot x_{ibr(t-1)}, \quad (10)$$

$$i = 1, 2, \dots, I, \quad r = 1, 2, \dots, R, \quad t = 2, 3, \dots, T, \quad b = 1, 2, \dots, B,$$

$$\sum_{i=1}^I W_i \cdot x_{ibrt} + \left(1 - \sum_{i=1}^I x_{ibrt} \right) M - \sum_{i=1}^I W_i \cdot x_{ibr(t-1)} + y_{br(t-1)} \cdot M \geq 0, \quad (11)$$

$$i = 1, 2, \dots, I, \quad r = 1, 2, \dots, R, \quad t = 2, 3, \dots, T, \quad b = 1, 2, \dots, B$$

$$x_{ibrt} \in \{0, 1\}, \quad i = 1, 2, \dots, I, \quad b = 1, 2, \dots, B, \quad r = 1, 2, \dots, R, \quad t = 1, \dots, T \quad (12)$$

$$y_{br(t-1)} \in \{0, 1\}, \quad b = 1, 2, \dots, B, \quad r = 1, 2, \dots, R, \quad t = 2, \dots, T$$

$$Z_b \geq 0, \quad b = 1, 2, \dots, B \quad (13)$$

Objective (1) is for the minimization of the travel distance of internal truck. Objective (2) is to minimize the number of reshuffles. Objective (3) indicates that the number of containers among bays needs to be balanced. The global objective is normalized using min-max normalization method and is shown as equation (4). Constraint (5) ensures that one container can only be located at one location exactly. Constraint (6) guarantees that no container can stack above an empty

location in the same stack. Constraint (7) indicates that one location can be occupied by no more than one container. Constraint (8) states that the number of containers does not exceed the capacity of the bay. Constraint (9) defines the workload in each bay. Constraint (10) ensures that the stacking order of outbound containers above the second floor should be no less than that of outbound containers below it. Constraint (11) represents the constraint condition satisfied

by the decision variable $y_{br(t-1)}$. Constraint (12) specifies the binary decision variables. Constraint (13) guarantees the nonnegative values of the variables.

4. The Proposed Algorithm

4.1. The Basic Principle of the Proposed Algorithm. Particle swarm optimization (PSO), derived from the study of bird flocking, is a population-based stochastic search method proposed by Kennedy et al. [43]. For the D-dimension problem, PSO algorithm maintains m particles and each particle i ($1 \leq i \leq m$) has a position vector $x_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ and a velocity vector $v_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$. The historical best position vector of particle i is $pbest_i = [pbest_{i1}, pbest_{i2}, \dots, pbest_{iD}]$ and the best position vector of all particles is $gbest = [gbest_1, gbest_2, \dots, gbest_D]$. In PSO, a candidate solution is represented as a particle. And each particle adapts its search patterns combined self-experiences and social experiences. Moreover, the vectors x_{iD}^{k+1} and v_{iD}^{k+1} are updated using the following formula:

$$v_{iD}^{k+1} = \omega v_{iD}^k + c_1 rand_1 (pbest_{iD}^k - x_{iD}^k) + c_2 rand_2 (gbest_D^k - x_{iD}^k) \quad (14)$$

$$x_{iD}^{k+1} = x_{iD}^k + v_{iD}^{k+1} \quad (15)$$

where ω is the inertia weight which decides convergence speed of PSO. Coefficients c_1 and c_2 are acceleration constants named the cognitive and social parameter, respectively. $rand_1$ and $rand_2$ represent two randomly generated numbers in the range $[0, 1]$.

As is well known, classical PSO is prone to premature convergence for the lack of population diversity [44, 45]. Hence, this paper proposes a neighborhood-based mutation operator strategy to enhance the diversity of the population and improve exploitation ability. However, it is prone to trap into local optima, if the radius of neighborhood is too small. Therefore, an intermediate disturbance strategy is introduced to help particle swarm jumping out of local optima by a greater probability and better achieve the global optimal solution.

4.2. Neighborhood-Based Mutation Operator. Based on the idea of differential evolution (DE), a neighborhood-based mutation operator strategy is obtained and introduced into PSO algorithm in order to achieve the purpose of enhancing population diversity, in which the local neighborhood of each particle could be found through a circular topology.

In Figure 3, the ring topology of neighborhood is depicted, where there are 16 particles in the swarm and the radius of circular topology is 2. Radius R is a nonzero integer, which ranges from 0 to $(m-1)/2$. According to the previous best particle $pbest$ and two other particles chosen from the R -neighborhood radius of x_i , the neighborhood-based mutation operator model is defined as follows:

$$X_{iD}^{k+1} = x_{iD}^k + r_{i,1} \cdot pbest_{iD}^k + r_{i,2} \cdot (X_{i_1D}^k - X_{i_2D}^k) \quad (16)$$

FIGURE 3: Neighborhood of X_i and search area of X_{iD}^{k+1} .

where X_{i_1} and X_{i_2} are two different particles produced by the i th particle X_i through the neighborhood radius R ; $r_{i,1}$ and $r_{i,2}$ are random numbers in $[0, 1]$ and $r_{i,1} + r_{i,2} = 1$, which are generated anew in each generation. The search area of X_{iD}^{k+1} is described in Figure 3. It is clear that $pbest_{iD}^k$ is the previous best particle of X_{iD}^{k+1} , so $pbest_{iD}^k$ is not on the circle topology.

When the optimal solution obtained by using this strategy is close to the global optimal solution, it displays that the algorithm has a smaller jump and better stability. However, if the radius of the particle is not properly selected, it is easy to fall into the local optimal solution. Therefore, it is extremely necessary to introduce an intermediate disturbance strategy to change the position of the particles so that they could quickly jump out of the local solution, thereby obtaining the global optimal solution.

4.3. Intermediate Disturbance Strategy. To conquer the premature convergence of PSO and to accelerate the convergence of particles, an intermediate disturbance strategy is introduced. The intermediate disturbance strategy was proposed by Grime [46] and developed by Connell in 1978 [47]. Existent literature indicates that if particles are given some disturbances in each iteration, the ability of particle swarms to search the global solution could be greatly improved. Therefore, the position formula of the constructed particle is constructed as follows:

$$y_{iD}^{k+1} = v_{iD}^k + \alpha \cdot abs(\beta) \cdot mbest \quad (17)$$

$$\beta = \frac{normrnd(0, \gamma)}{(rand)} \quad (18)$$

$$mbest = \frac{\sum_{i=1}^m X_{iD}^k}{m} \quad (19)$$

In Eq. (18), the value of γ is set to 0.35 and in Eq. (17) α is set to 0.5, which have been proved to be the best choice in [41]. According to the intermediate disturbance theory, a larger disturbance would lead to chaos of particles while a smaller one would make the contribution of PSO weak in global search capability. Hence, the setting of values α and γ is critical to the new PSO algorithm.

TABLE 1: A coding example.

Container number	1	2	3	4	5	6	7
Stack number	1	2	3	3	2	1	3
Particle value	0.82	1.56	2.44	2.71	1.32	0.38	2.74

4.4. Encoding Representation and Initial Solution. Coding is the key factor affecting PSO algorithm. It could be a string of integers or real numbers. Considering the NP-hard combinatorial nature of OCSLAP, stack-based integer encoding representation is adopted in discrete search space. In this encoding scheme, a solution for the problem of assigning containers to locations is represented by an array which length equals the number of containers.

Take seven containers, three stacks, and three tiers, for instance. In Table 1, the first line represents the number of incoming containers, the second line a solution $x = [1, 2, 3, 3, 2, 1, 3]$, and the third line the value of particles randomly generated in the range $[1, 3]$. The solution x indicates that the first and the sixth containers are allocated to stack 1, the second and the fifth containers are allocated to stack 2, and the third, fourth, and seventh containers are allocated to stack 3. The stack number is obtained by a minimum integer conversion that is not less than the particle value itself. For instance, the particle value of the fifth container is 1.32, and the smallest integer more than 1.32 is 2. So, the fifth container is assigned to stack 2.

In this article, initial solution is generated randomly. However, there exists a series of infeasible solution, such as $x = [1, 2, 2, 3, 2, 1, 2]$, where the total number of containers assigned to stack 2 exceeds 3. Infeasible solution will reduce the quality of population, so repairing strategy [48] is adopted to improve initial solution to obtain the high-quality initial solution.

In reality, the lowest stack principle [49] is applied to ensure the stability of stacking in container yard. This article incorporates the lowest stack principle into repairing strategy so that the feasible solution obtained is consistent with the actual situation. Combined with stack-based coding, the lowest stack principle means that the number of containers assigned to the specified stack is minimal. Take the infeasible solution $x = [1, 2, 2, 3, 2, 1, 2]$ as an example. The total number of containers assigned to stacks 1, 2, and 3 are 2, 4, and 1, respectively. Obviously, the total number of containers assigned to stack 2 exceeds 3. So, it is necessary to use the lowest stack principle to adjust the infeasible solution. Select any container assigned to stack 2 and assign it to stack 3 because the total number of containers assigned to stack 3 is the least. The feasible solution gained is $x = [1, 2, 3, 3, 2, 1, 2]$.

4.5. Procedure of the Proposed Algorithm. The flowchart of the proposed algorithm is presented in Figure 4. The detailed steps are as follows:

Step 1 (initialization). Set parameters such as c_1, c_2, m, ω, k and $Maxiter$. Initialize the particle position x_i and particle v_i randomly.

FIGURE 4: The flowchart of the proposed algorithm.

Step 2 (treatment of infeasible solutions). Infeasible solutions are adjusted by repairing strategy.

Step 3 (fitness calculation). Compute fitness function (4) and find $pbest_i$ and $gbest$.

Step 4 (particle velocity construction). The particle velocity v_i is defined through formula (14).

TABLE 2: Benchmark optimization functions.

Name	Function formula	Domain	$f_{\min}$
Rosenbrock	$f_1 = \sum_{i=1}^D [100(x_i^2 - x_{i+1}^2)^2 + (x_i - 1)^2]$	$[-2.048, 2.048]$	0
Ackley	$f_2 = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]$	0
Griewank	$f_3 = \sum_{i=1}^D \frac{x_i^2}{40000} - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]$	0
Alpine	$f_4 = \sum_{i=1}^D x_i \sin(x_i) + 0.1 x_i $	$[-10, 10]$	0
Rastrigin	$f_5 = \sum_{i=1}^D (x_i^2 - 10 * \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0

TABLE 3: The mean and standard deviations of the function values for 30-D problems.

Function		TPA	CPSO-H ₆	LPSO	FIPS	CLPSO	APSO
f_1	Mean	<i>4.39e+00</i>	3.23e+03	6.71e+01	3.32e+01	2.76e+01	3.05e+02
	Std	<i>4.89e+00</i>	1.64e+04	4.00e+01	2.12e+01	2.14e+01	5.29e+02
f_2	Mean	<i>1.89e-09</i>	4.81e-05	1.62e-05	5.41e-07	2.95e-09	9.01e-02
	Std	<i>1.04e-09</i>	3.09e-05	1.18e-05	2.32e-07	1.12e-09	4.98e-02
f_3	Mean	<i>1.72e-09</i>	8.06e-02	1.79e-02	4.77e-04	4.27e-09	3.38e-01
	Std	<i>2.62e-09</i>	5.91e-02	1.55e-02	1.95e-03	1.41e-08	2.35e-01
f_4	Mean	<i>3.89e-05</i>	1.65e-01	2.22e-03	6.64e-03	5.99e-04	8.51e-03
	Std	<i>3.42e-05</i>	8.08e-01	4.06e-03	1.63e-03	3.63e-04	9.75e-03
f_5	Mean	<i>2.19e-09</i>	5.25e+01	4.46e+01	8.45e+01	3.32e-02	1.17e+00
	Std	<i>2.40e-09</i>	2.08e+01	1.31e+01	1.16e+01	1.85e-01	8.61e-01

Note: TPA stands for the proposed algorithm. “Mean” is the mean of the solutions. “Standard deviation” is abbreviated as “Std”. The numbers in italic denote the best solution.

Step 5 (neighborhood-based mutation operator execution). Calculate X_{iD}^{k+1} by formula (16) to enhance the population diversity.

Step 6 (intermediate disturbance strategy execution). Define the particle position y_i using formula (17).

Step 7 (update $pbest_i$ and $gbest$). First, keep the boundary value of the particle position; then perform repairing strategy and compute the fitness values; finally, update the personal best $pbest_i$ and the global best $gbest$.

Step 8 (termination). If it satisfies the stop criterion, the algorithm terminates; otherwise go to Step 4.

5. Numerical Experiments

5.1. Test Functions. In order to assess the performance of the proposed algorithm, a series of benchmark functions are tested, as detailed in Table 2. Five existing PSO algorithms are used for the comparison, which are CPSO-H_k [50], LPSO [51], FIPS [52], CLPSO [53], and APSO [54], respectively. In the evaluation, to fairly compare the proposed algorithm with other methods, each algorithm will run independently

30 times. The dimension D is set to 30 and the population size m in each algorithm is set to 20. The inertia weight ω linearly decreases from 0.9 to 0.4 and c_1 and c_2 are initialized to 2.

The results are displayed in Table 3 in terms of the mean and standard deviation of the solutions. And the best results are marked in italic. For the five functions ($f_1 - f_5$), the results show that the proposed algorithm achieves better solutions than other peer algorithms, which means that the proposed PSO algorithm achieves a balance between exploration and exploitation searches through combining the neighborhood-based mutation operator strategy and the intermediate disturbance strategy. Therefore, in terms of the obtained mean and standard deviation, the proposed algorithm is effective.

5.2. Parameters Setting. In metaheuristic algorithms, parameter setting is necessary, for it is a critical factor that matters to the convergence and stability of algorithms. The control parameters of PSO are inertia weight ω , the number of particles m , self-learning factors c_1 , social-learning factors c_2 , and a number of iterations $Maxiter$, respectively. To get best values, a series of frequently used experimental parameters are tested and good results are available in most

TABLE 4: Results of cases with different parameters.

Scenario	Parameters				Case 1-2 × 5 × 4 (30)			Case 2-4 × 6 × 4 (72)		
	m	c_1	c_2	$Maxiter$	Avg	Opt	Std	Avg	Opt	Std
1	50	1	2	200	0.05124	0.0333	0.0045	0.0888	0.0722	0.0053
2	50	2	2	200	0.05078	0.0333	0.0042	0.0881	0.0722	0.0051
3	50	4	1	200	0.05147	0.0333	0.0047	0.0905	0.0722	0.0053
4	50	2	2	100	0.05317	0.0333	0.0054	0.0933	0.0722	0.0055
5	100	2	2	200	0.04997	0.0333	0.0042	0.0852	0.0722	0.0049
6	100	2	2	100	0.05194	0.0467	0.0039	0.0917	0.0806	0.0053
7	100	1	4	200	0.05066	0.0333	0.0043	0.0875	0.0722	0.0050

FIGURE 5: The average values and standard deviation of different scenarios.

instances, even though the parameters are not the optimal setting for all instances. Each instance is run for 30 times. All experiments are run in Windows 10 on a desktop PC with Intel Core i3-4170 CPU, 3.7 GHz processor, and 4.00 GB memory.

Since the larger weight is conducive to jumping out of the local minimum value and facilitating global search and the smaller weight is more conducive to strengthening the local search of the current region and facilitating the convergence of the algorithm, this study adopts the linear weight ($\omega = \omega_{max} - i * (\omega_{max} - \omega_{min}) / Maxiter$, $\omega_{max} = 0.9$, $\omega_{min} = 0.4$) to balance the global search capability and local exploitation ability. Other parameters used in the experiments are listed in Table 2. Seven scenarios with each of 420 instances are solved by the proposed algorithm. Column 1 represents different scenarios, columns 2-5 represent different parameters including m , c_1 , c_2 and $Maxiter$, and the remaining columns represent the average values (Avg), the optimal values (Opt), and standard deviation (Std) generated by different cases. It can be observed from Table 4 and Figure 5 that the results produced in the fifth scenario are the best in terms of the average values, the optimal values, and standard deviation. So, in the following section, parameters are set as follows: $c_1 = c_2 = 2$, $\omega = 0.9 - i * 0.5 / Maxiter$, $m = 100$, and $Maxiter = 200$.

5.3. Comparative Analysis. In this part, the proposed method is applied to solve a real case, which comes from a small and medium terminal in China. According to the actual investigation of Guangdong-Hong Kong-Macao Greater Bay

Area, this paper sets the max stack height to 6 in one bay, and a series of small and medium-sized cases are tested. The results are displayed in Table 5 and Figures 6 and 7.

In Table 5, column 1 indicates bay×stack×tier and the number of containers, columns 2-4 indicate different algorithms including TPA, PSO, and GA, and columns 5-6 indicate improved efficiency. The numbers in italic denote the best values. It can be observed that the proposed algorithm is superior to other two approaches in terms of the optimal solution. At the same time, compared with the traditional PSO, the results gained by the proposed algorithm are greatly improved, about 38% less than the previous results and about 58% less than results by GA. In addition, it can be seen from Figure 6 that the average values and standard deviations gained by the proposed algorithm are smaller than those of the other two algorithms, demonstrating that the proposed algorithm has a better effect and stability than PSO and GA. But, in Figure 7, it is clear that the proposed algorithm consumes more time than PSO and GA regardless of the number of containers. The reason for this is that, to make sure that the efficient solution is obtained, the lowest stack principle is executed twice during Matlab program. The above results indicate that the proposed algorithm could be used to solve OCSLAP of small and medium terminals.

5.4. Discussion. Comparative trend graphs of convergence of both general PSO and proposed algorithm with 200 simulation iterations are displayed in Figure 8. Both curves follow a

TABLE 5: Comparison of the results of different approaches.

Bay×stack×tier (containers)	Different algorithms			Improved efficiency	
	TPA	PSO	GA	(1-TPA/PSO)×100%	(1-TPA/GA)×100%
1 × 4 × 4 (14)	<i>0.0417</i>	0.0617	0.1333	32.42%	68.72%
1 × 6 × 4 (20)	<i>0.0556</i>	0.1333	0.1667	58.29%	66.41%
1 × 10 × 6 (55)	<i>0.0833</i>	0.1237	0.1534	32.66%	45.70%
2 × 4 × 3 (22)	<i>0.0417</i>	0.0542	0.0752	23.06%	44.55%
2 × 5 × 4 (30)	<i>0.0333</i>	0.0833	0.15	60.02%	77.8%
2 × 6 × 4 (36)	<i>0.0278</i>	0.0694	0.125	59.94%	77.76%
2 × 6 × 4 (44)	<i>0.0694</i>	0.083	0.132	16.39%	47.42%
2 × 6 × 6 (65)	<i>0.0778</i>	0.0944	0.1748	17.58%	55.49%
2 × 6 × 6 (70)	<i>0.1222</i>	0.1561	0.191	21.72%	36.02%
4 × 6 × 4 (72)	<i>0.0722</i>	0.1444	0.1722	50%	58.07%
4 × 6 × 4 (90)	<i>0.1132</i>	0.1723	0.2615	34.31%	56.71%
7 × 5 × 4 (105)	<i>0.0857</i>	0.1567	0.327	45.31%	73.79%
7 × 6 × 3 (110)	<i>0.0884</i>	0.1484	0.1924	40.43%	54.05%
7 × 6 × 3 (120)	<i>0.1087</i>	0.1737	0.197	37.42%	44.82%
7 × 6 × 4 (126)	<i>0.0509</i>	0.1071	0.159	52.47%	67.99%
Average				38.8%	58.35%

Note: TPA stands for the proposed algorithm. The best values are marked in italic.

FIGURE 6: The average values and standard deviations obtained using different methods.

FIGURE 7: Computational time consumed by different approaches.

TABLE 6: Experiments results under different scenarios.

Bay×stack×tier (containers)	f_1	f_2	f_3	f
$2 \times 4 \times 3$ (22)	33	1	0	0.0417
$2 \times 6 \times 4$ (36)	54	2	0	0.0278
$2 \times 6 \times 4$ (44)	66	5	0	0.0694
$2 \times 6 \times 6$ (65)	97	14	1	0.0778
$2 \times 6 \times 6$ (70)	105	22	0	0.1222
$4 \times 6 \times 4$ (72)	177	13	2	0.0722
$4 \times 6 \times 4$ (90)	219	25	3	0.1132
$7 \times 6 \times 3$ (110)	416	24	4	0.0884
$7 \times 6 \times 3$ (120)	468	35	2	0.1087

FIGURE 8: Convergence characteristics of PSO and the proposed algorithm.

downward trend, in which PSO and the proposed algorithm reach the optimal solution when it is 168 and 105 iterations, respectively, which means that the proposed algorithm has relatively good convergence ability. The main reason of this phenomenon is that when the particle swarm begins to fall into the local solution, the intermediate disturbance strategy is applied to the particle swarm, and the particle could quickly jump out of the local solution with a certain probability which results in finding the global optimal solution with fewer iterations. In addition, the repairing strategy is prone to a sudden drop in the curve. In summary, the proposed algorithm is able to balance the exploitation and exploration capabilities and is an effective algorithm.

A set of experiments employing the proposed algorithm are performed and the detailed results are presented in Table 6. As expected, the value of f_3 is relatively small because the number of containers in the bay can be adjusted through the lowest stack strategy. For instance, for the “ $7 \times 6 \times 3$ (110)” scenario, the imbalance of the number of containers among 7 bays is 4. Another important observation is that, for the same bay size, the more the number of containers, the larger the values of f_1 and f_2 . For the “ $2 \times 6 \times 6$ (65)” and “ $2 \times 6 \times 6$ (70)” problem scale, the values of f_2 (i.e., the number of reshuffles) are 14 and 22, respectively. There is little difference in the number of containers, while there is a big difference in the number of reshuffles, which may be caused by higher bay

utilization rate. In other words, the higher the bay utilization rate, the more the number of reshuffles. Hence, it is necessary to choose the appropriate bay utilization rate to reduce the number of reshuffles when solving OCSLAP.

6. Conclusions

In this paper, the proposed PSO algorithm is applied to solve OCSLAP. Computational results indicate that, in the small and medium scale, the target function value obtained by the proposed algorithm is much smaller than PSO and GA and its improvement efficiency is 38% and 58%, respectively. In addition, results also prove that the proposed algorithm could solve combinatorial problems well.

This study could be further extended from the following perspectives. First, a new intelligent algorithm that is tree-seed algorithm (TSA) could be considered to solve OCSLAP, because the TSA has two different search equations and, for each tree, more than one seed (candidate solution) is created and these equations and seed production mechanism could provide balanced local and global search capability [55]. Second, some uncertain factors such as the uncertainty of container weight level information and outbound container arrival time could be considered when solving OCSLAP so that it is consistent with the practical operations.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the 2017 Founded Major Project of NSFC (71731006) and Ministry of Education (MOE) of China Project of Key Research Institute of Humanities and Social Sciences at Universities (17JZD020).

References

- [1] F. Bruns, S. Knust, and N. V. Shakhlevich, "Complexity results for storage loading problems with stacking constraints," *European Journal of Operational Research*, vol. 249, no. 3, pp. 1074–1081, 2016.
- [2] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, "Hierarchical genetic algorithm with new evaluation function and bi-coded representation for the selection of features considering their confidence rate," *Applied Soft Computing*, vol. 11, no. 2, pp. 2501–2509, 2011.
- [3] M. A. Salido, J. Escamilla, A. Giret, and F. Barber, "A genetic algorithm for energy-efficiency in job-shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5–8, pp. 1304–1314, 2016.
- [4] S. Tabakhi, A. Najafi, R. Ranjbar, and P. Moradi, "Gene selection for microarray data classification using a novel ant colony optimization," *Neurocomputing*, vol. 168, no. 30, pp. 1024–1036, 2015.
- [5] M. Danesh and H. Shirgahi, "A novel hybrid knowledge of firefly and pso swarm intelligence algorithms for efficient data clustering," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 33, no. 6, pp. 1–10, 2017.
- [6] Z. H. Huang and Y. D. Chen, "Log-linear model based behavior selection method for artificial fish swarm algorithm," *Computational Intelligence and Neuroscience*, vol. 2015, no. 1, Article ID 685404, 10 pages, 2015.
- [7] C.-J. Ting, K.-C. Wu, and H. Chou, "Particle swarm optimization algorithm for the berth allocation problem," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1543–1550, 2014.
- [8] X. Han, X. Gong, and J. Jo, "A new continuous berth allocation and quay crane assignment model in container terminal," *Computers & Industrial Engineering*, vol. 89, no. 4, pp. 15–22, 2015.
- [9] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 58–73, 2002.
- [10] M. C. Chih, "Three pseudo-utility ratio-inspired particle swarm optimization with local search for multidimensional knapsack problem," *Swarm and Evolutionary Computation*, vol. 39, pp. 279–296, 2017.
- [11] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing*, vol. 43, pp. 117–130, 2016.
- [12] M. M. Javidi and N. Emami, "A hybrid search method of wrapper feature selection by chaos particle swarm optimization and local search," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24, no. 5, pp. 3852–3861, 2016.
- [13] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.
- [14] K. H. Kim, Y. M. Park, and K. Ryu, "Deriving decision rules to locate export containers in container yards," *European Journal of Operational Research*, vol. 124, no. 1, pp. 89–101, 2000.
- [15] C. R. Zhang, W. W. Chen, L. Y. Shi, and L. Zheng, "A note on deriving decision rules to locate export containers in container yards," *European Journal of Operational Research*, vol. 205, no. 2, pp. 483–485, 2010.
- [16] L. Chen and Z. Q. Lu, "The storage location assignment problem for outbound containers in a maritime terminal," *International Journal of Production Economics*, vol. 135, no. 1, pp. 73–80, 2012.
- [17] R. Guerra-Olivares, R. G. González-Ramírez, and N. R. Smith, "A heuristic procedure for the outbound container relocation problem during export loading operations," *Mathematical Problems in Engineering*, vol. 2015, Article ID 201749, 13 pages, 2015.
- [18] W. B. Hu, W. Huang, and Z. Y. Min, "A storage allocation algorithm for outbound containers based on the outer-inner cellular automaton," *Information Sciences*, vol. 281, pp. 147–171, 2014.
- [19] Y. Marinakis and M. Marinaki, "A hybrid genetic - particle swarm optimization algorithm for the vehicle routing problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1446–1455, 2010.
- [20] A. H. Kashan and B. Karimi, "A discrete particle swarm optimization algorithm for scheduling parallel machines," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 216–223, 2009.
- [21] T. Ren, P. Zhao, D. Zhang, B. Q. Liu, and H. W. Yuan, "Permutation flow-shop scheduling problem to optimize a quadratic objective function," *Engineering Optimization*, vol. 49, no. 9, pp. 1589–1603, 2017.
- [22] B. Niu, T. Xie, L. J. Tan, Y. Bi, and Z. X. Wang, "Swarm intelligence algorithms for yard truck scheduling and storage allocation problems," *Neurocomputing*, vol. 188, pp. 284–293, 2016.
- [23] M. J. Xia, N. Zhao, and W. J. Mi, "Storage allocation in automated container terminals: the upper level," *Polish Maritime Research*, vol. 23, no. s1, pp. 160–174, 2016.
- [24] N. Zhao, M. J. Xia, C. Mi, Z. C. Bian, and J. Jin, "Simulation-based optimization for storage allocation problem of outbound containers in automated container terminals," *Mathematical Problems in Engineering*, vol. 2015, no. 1, Article ID 548762, 14 pages, 2015.
- [25] Y. J. Zhang, Y. Zhao, X. H. Fu, and J. R. Xu, "A feature extraction method of the particle swarm optimization algorithm based on adaptive inertia weight and chaos optimization for Brillouin scattering spectra," *Optics Communications*, vol. 376, pp. 56–66, 2016.
- [26] E. Zahara and Y.-T. Kao, "Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3880–3886, 2009.
- [27] J. I. Ding, J. Liu, K. R. Chowdhury, W. S. Zhang, Q. P. Hu, and J. Lei, "A particle swarm optimization using local stochastic search and enhancing diversity for continuous optimization," *Neurocomputing*, vol. 137, no. SI, pp. 261–267, 2014.
- [28] Ş. Gülcü and H. Kodaz, "A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 33–45, 2015.
- [29] D. Tian and Z. Shi, "MPSO: Modified particle swarm optimization and its applications," *Swarm and Evolutionary Computation*, vol. 41, pp. 49–68, 2018.
- [30] B. Farnad, A. Jafarian, and D. Baleanu, "A new hybrid algorithm for continuous optimization problem," *Applied Mathematical Modelling*, vol. 55, pp. 652–673, 2018.
- [31] Y. G. Chen, L. X. Li, J. H. Xiao, Y. X. Yang, J. Liang, and T. Li, "Particle swarm optimizer with crossover operation," *Engineering Applications of Artificial Intelligence*, vol. 70, pp. 159–169, 2018.

- [32] J. T. Lin and C. Chiu, "A hybrid particle swarm optimization with local search for stochastic resource allocation problem," *Journal of Intelligent Manufacturing*, vol. 29, no. 3, pp. 481–495, 2018.
- [33] F. Wang, H. Zhang, K. S. Li, Z. Y. Lin, Y. Yang, and X. L. Shen, "A hybrid particle swarm optimization algorithm using adaptive learning strategy," *Information Sciences*, vol. 436/437, pp. 162–177, 2018.
- [34] Y. M. Liu, B. Niu, and Y. H. Luo, "Hybrid learning particle swarm optimizer with genetic disturbance," *Neurocomputing*, vol. 151, no. 3, pp. 1237–1247, 2015.
- [35] H. Wang, W. J. Wang, and Z. J. Wu, "Particle swarm optimization with adaptive mutation for multimodal optimization," *Applied Mathematics and Computation*, vol. 221, pp. 296–305, 2013.
- [36] Q. Wu, "Hybrid forecasting model based on support vector machine and particle swarm optimization with adaptive and Cauchy mutation," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9070–9075, 2011.
- [37] H. T. Liang and F. H. Kang, "Adaptive mutation particle swarm algorithm with dynamic nonlinear changed inertia weight," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 19, pp. 8036–8042, 2016.
- [38] P. Bia, D. Caratelli, L. Mescia, and J. Gielis, "Analysis and synthesis of supershaped dielectric lens antennas," *IET Microwaves, Antennas & Propagation*, vol. 9, no. 14, pp. 1497–1504, 2015.
- [39] H. Lu, P. Sriyanyong, Y. H. Song, and T. Dillon, "Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 9, pp. 921–935, 2010.
- [40] J. J. Chen, Y. S. Ding, and K. Hao, "The bidirectional optimization of carbon fiber production by neural network with a GA-IPSO hybrid algorithm," *Mathematical Problems in Engineering*, vol. 2013, no. 6, Article ID 768756, 16 pages, 2013.
- [41] H. Gao, S. Kwong, J. Yang, and J. Cao, "Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation," *Information Sciences*, vol. 250, pp. 82–112, 2013.
- [42] L. Mescia, P. Bia, D. Caratelli, and J. Gielis, "Swarm intelligence for electromagnetic problem solving," in *Handbook of Research on Soft Computing and Nature-Inspired Algorithms*, S. K. Shandilya, S. Shandilya, K. Deep, and A. K. Naga, Eds., Advances in Computational Intelligence and Robotics, pp. 69–100, IGI Global, Hershey, Pa, USA, 2017.
- [43] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks (Perth)*, pp. 1942–1948, IEEE Service Center, Piscataway, NJ, USA, 1995.
- [44] W. X. Ye, W. Y. Feng, and S. H. Fan, "A novel multi-swarm particle swarm optimization with dynamic learning strategy," *Applied Soft Computing*, vol. 61, pp. 832–843, 2017.
- [45] M. S. Kiran, "Particle swarm optimization with a new update mechanism," *Applied Soft Computing*, vol. 60, pp. 670–678, 2017.
- [46] J. P. Grime, "Competitive exclusion in herbaceous vegetation," *Nature*, vol. 242, no. 5396, pp. 344–347, 1973.
- [47] J. H. Connell, "Diversity in tropical rain forests and coral reefs," *Science*, vol. 199, no. 4335, pp. 1302–1310, 1978.
- [48] X. Hu and R. Eberhart, "Solving constrained nonlinear optimization problems with particle swarm optimization," in *Proceedings of the 6th World Multi-conference on Systemics, Cybernetics and Informatics*, pp. 203–206, ORLANDO, Florida, Fla, USA, 2002.
- [49] M. J. Ji, W. W. Guo, H. L. Zhu, and Y. Z. Yang, "Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals," *Transportation Research Part E: Logistics and Transportation Review*, vol. 80, pp. 1–19, 2015.
- [50] F. V. D. Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [51] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1671–1676, Honolulu, Hawaii, USA, May 2002.
- [52] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [53] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [54] Z.-H. Zhan, J. Zhang, and Y. Li, "Adaptive particle swarm optimization," *IEEE Transactions on Systems Man and Cybernetics Part B- Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [55] A. Babalik, A. C. Cinar, and M. S. Kiran, "A modification of tree-seed algorithm using Deb's rules for constrained optimization," *Applied Soft Computing*, vol. 41, pp. 599–609, 2017.