Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6352539, 12 pages
https://doi.org/10.1155/2020/6352539

Research Article

Optimization of Order-Picking Problems by Intelligent Optimization Algorithm

Zhong-huan Wu,[1] **Hong-jie Chen**,[2] **and Jia-jia Yang**[3]

[1]*Department of Business Administration, Huashang College Guangdong University of Finance & Economics, Guangzhou 510000, China*
[2]*School of Business Administration, South China University of Technology, Guangzhou 510000, China*
[3]*Department of Public Policy, King's College London, London WC2R 2LS, UK*

Correspondence should be addressed to Hong-jie Chen; raphel888@gmail.com

Received 29 April 2020; Accepted 3 June 2020; Published 23 July 2020

Guest Editor: Wen-Tsao Pan

To improve the efficiency of warehouse operations, reasonable optimization of picking operations has become an important task of the modern supply chain. For the purpose of optimization of order picking in warehouses, a new fruit fly optimization algorithm, particle swarm optimization, random weight, and weight decrease model are used to solve the mathematical model. Further optimization is achieved through the analysis of the warehouse shelves and screening of the optimal solution of the picking time. In addition, simulation experiments are conducted in the MATLAB environment through programming. The shortest picking time is found out and chosen as an optimized method by taking advantage of the effectiveness of these six algorithms in the picking optimization and comparing the data obtained under the simulation. The result shows that the optimization capacity of RWFOA is better and the picking efficiency is the best.

1. Introduction

As a part of the logistics, the efficiency of the automated warehouse is largely dependent on the efficiency of order picking. Therefore, the picking plays an important role in the automated warehouse for improving the efficiency of picking operation. Although the automated stereoscopic warehouse provides more orderly and standardized management and the error rate is also small, for small batch warehouse, frequent warehousing, and warehouses with various products, logistics storage becomes more stringent, and requirements for the efficiency of the logistics are higher, and thus, the efficiency of picking needs to be improved.

There are a lot of algorithms for optimization of order picking, all of which have made minor or major contributions to the optimization of order picking [1–6]. At present, the ant colony algorithm [7], genetic algorithm [8] and multipopulation fruit fly optimization algorithm [9], which have been used to solve the picking operation problems, yielded good results. Based on the existing research, we will use the new fruit fly optimization algorithm and particle swarm optimization to solve the mathematical model.

The main structure of this paper is as follows: Section 1 introduces the motive and purpose of this study; Section 2 presents the literature review; Section 3 introduces research methods—original FOA, original particle swarm algorithm (PSA), random weight algorithm, weight decrease, and related literature; Section 4 introduces case description; Section 5 presents results and discussion; and Section 6 puts forward the research conclusions and suggestions.

2. Literature Review

2.1. Order Picking. Across the various operations in a warehouse, order picking is the most time-consuming operation in general [10] and accounts for around 55–75 percent of total warehousing costs [11]. Therefore, order

picking has the highest priority for productivity improvement [12].

Order picking is a particular case of the traveling salesman problem (TSP). This problem, introduced by Dantzig et al. [13], is one of the most studied problems in operations research. Efficient algorithms have been designed for the TSP [14]. Therefore, in order to improve the performance of order picking, reducing travel time is critical. Since the travel distance is proportional to travel time for picker-to-parts system [15], minimizing the travel distance (total or average) of a picking tour is often considered as an imperative factor to reduce travel time and consequently improve warehouse operation efficiency [12]. There are four methods to reduce the travel distance of an order picker [12]: storage location assignment, warehouse zoning, order batching, and pick-routing methods. And this paper focuses on the pick-routing methods.

To most order-picking research studies, optimization algorithms are still the center of routing studies [16]. To pursue the optimal order-picking route in a typical rectangular, the order-picking routing problem is considered as the STSP (Steiner traveling salesman problem) [12, 17]. There are two general methods to solve the STSP: the first method is to reformulate an STSP into the classic TSP by computing the shortest paths between every pair of required nodes (e.g., Renaud and Ruiz [17]) and the second one for the solution of a STSP is by using dedicated algorithms (e.g., Lucie Pansart [4]). The latter method is preferred to the former.

The dedicated algorithms include dynamic programming, integer programming, and branch and bound method. Although this kind of algorithm can get the exact solution, the calculation time is long and it is seldom used in the practical application [18]. The common approximation algorithms are the insertion algorithm, the r-opt algorithm, and the nearest neighbour algorithm. Although this kind of algorithm can quickly get a feasible solution to the optimal solution, the degree of its close to the optimal solution is not satisfactory [19]. Intelligent optimization algorithm is a more effective algorithm to solve this problem in recent years [20]. These algorithms are mainly genetic algorithms [8], ant colony algorithm [7], particle swarm optimization [21], and modified FOA, e.g., MSFOA [22] and IFOA4WSC [23].

2.2. Automated Stereoscopic Warehouse Model. At present, the shelves of the automatic stereoscopic warehouse are mainly fixed shelves, and each row of shelves in the warehouse is equipped with a stacker, which is responsible for picking up a cargo on the shelf. This paper takes some of the shelves in the warehouse as the object of study. In an automated warehouse, the stacker enters from the entrance, performs order picking, and chooses goods according to the programmed procedure. Assuming that there are k cargo spaces on each shelf, the stacker can only get one cargo space for each picking, removes the cargo from the shelve and transports it to the exit, and then returns to the shelves to pick, and the above steps were repeated. As the order of picking is not the same, the time required for each picking is not the same. We need to set the optimal sorting order, so as to minimize the time on picking and to improve the efficiency of order picking.

This study refers to the high-level rack model designed by Professor Ning and Hu [9], and the formulas from (1) to (8) are also proposed by them [9]. The structure is shown in Figure 1.

Where, the position of the column x and tier y can be set to (i, j), the position at the entrance is set to $(0, 0)$, the length of the shelf is D, and the height of the shelf is G. Assuming a shelf has x columns and y tiers, the goods allocation is (Y, j) $(m = 1, 2, 3, \ldots, k)$, and position is (α, β), so

$$\alpha m = \frac{D \cdot i}{X} - \frac{D}{2X}, \tag{1}$$

$$\beta m = \frac{G(j-1)}{Y}. \tag{2}$$

Assuming that the velocity in the horizontal direction of the stacker is v_1 and the velocity in the vertical direction is v_2, and the time spent by two adjacent cargo spaces m (i, j) and $m+1$ (p, q) in the process of picking up the goods in the horizontal direction is t_1 and t_2; the operating equation used is as follows:

$$t_1 = \frac{|\alpha_m - \alpha_{m+1}|}{v_1} = \frac{|i-p|D}{X \cdot v_1}, \tag{3}$$

$$t_2 = \frac{|\beta_m - \beta_{m+1}|}{v_2} = \frac{|j-q|G}{Y \cdot v_2}. \tag{4}$$

Since the horizontal and vertical movements of the stacker occur at the same time, the time of operation at the adjacent cargo space is t_m, and the maximum value for running speed t_1 in the horizontal direction and the running speed t_2 in the vertical direction is given by

$$t_m = \max\{t_1, t_2\}. \tag{5}$$

Then, the k cargo positions are selected and the total running time T_z used by the stacker is given by

$$T_z = \sum_{m=0}^{n} t_m. \tag{6}$$

If the picking time spent by the stacker is the same each time, that is, t_s, then the running time of all the cargo spaces is T_s, as shown in the following equation:

$$T_s = \sum_{m=1}^{n} t_s = kt_s. \tag{7}$$

Thus, the total time T of k cargo spaces is as follows:

$$T = T_z + T_s. \tag{8}$$

Under above circumstances, we will ask for the total time of operation of the automated warehouse stacker and the minimum value T. Six intelligence algorithms including original particle swarm, particle swarm weight decrease, particle swarm random weight, original FOA, fruit fly weight decrease, and fruit fly random weight are used to evaluate the minimum value of T.

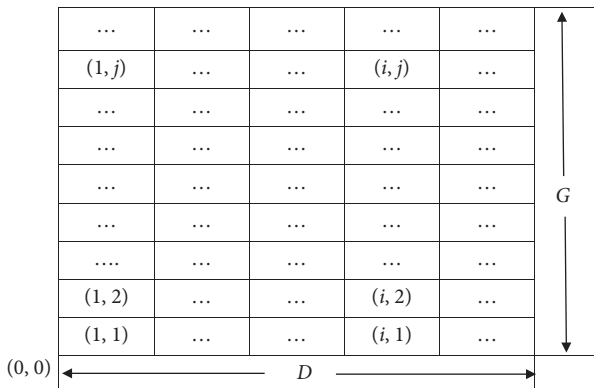


Figure 1: Structural model of high-level rack.

3. Research Methods

Particle swarm optimization is a new algorithm in recent years, which solves the TSP problem, and a good result is obtained [20]. And the fruit fly optimization algorithm (FOA) is a newly developed bioinspired algorithm. The continuous variant version of FOA has been proven to be a powerful evolutionary approach to determining the optima of a numerical function on a continuous definition domain [24]. The FOA and PSO are also easy to program and can be modified to other practical applications. Due to these advantages, they have been used to solve a wide range of optimization problems, including prediction and classification problems [25–27]. However, the FOA and PSO must be modified in order to effectively manage the discrete variables associated with optimization issues. Therefore, RW and WD were integrated into FOA and PSO to improve its advantage and to look for the better optimal order-picking time.

3.1. Fruit Fly Optimization Algorithm (FOA). The original FOA was invented by Professor Pan [24], and the FOA is highly accurate. Many studies will use the FOA to solve the optimization problem. FOA can be used in any field, such as military, engineering, medical science, management, and financial and other fields. It can also be combined with other algorithms, complementing each other. FOA is a new method of global optimization derived from foraging behaviors of fruit flies. Because a fruit fly itself is superior to other animals in perception, it comes close to the food using its olfactory organ, knowing where the food and partners gathered, and then fly to the destination. Following is the original fruit fly algorithm:

(1) Set initial location of fruit flies at random (x and y are two coordinate axes, initial position on coordinates):

$$\begin{aligned} &\text{Init}X \text{ axis}, \\ &\text{Init}Y \text{ axis}. \end{aligned} \tag{9}$$

(2) Random directions and distance of fruit flies searching for food relying on good sense of smell, which is equivalent to the initial location of the fruit flies plus random flight distance:

$$\begin{aligned} Xi &= X \text{ axis} + \text{Random Value}, \\ Yi &= Y \text{ axis} + \text{Random Value}. \end{aligned} \tag{10}$$

(3) As the location of food cannot be obtained, estimate the distance (Di) to the origin first, and then calculate the decision value of Smelli (Si), and this value is the reciprocal of Di:

$$\begin{aligned} Di &= \sqrt{X_i^2 + Y_i^2}, \\ Si &= \frac{1}{Di}. \end{aligned} \tag{11}$$

(4) Substitute decision value of Smelli (Si) into the above function to get the Smelli of location of fruit flies:

$$\text{Smelli} = \text{Function}(Si). \tag{12}$$

(5) Locate the fruit fly with the best Smelli from fruit flies (max):

$$[\text{bestSmellbestIndex}] = \max(\text{Smelli}). \tag{13}$$

(6) Retain the smell best and X-axis and Y-axis, and the fruit flies will fly to this position.

$$\begin{aligned} \text{Smellbest} &= \text{bestSmell}, \\ X \text{ axis} &= X(\text{bestIndex}), \\ Y \text{ axis} &= Y(\text{bestIndex}). \end{aligned} \tag{14}$$

(7) Enter into iterative optimization, repeat steps 2–5, and judge whether the Smelli is superior to the Smelli of the previous iteration, if yes, execute step 6.

The foraging process of a fruit fly group is illustrated in Figure 2 [25].

In view of the optimization of picking in this paper, we know that the range of search distance of the original fruit fly in the coordinates is limited, which leads to the weak optimal performance. If the weight is added to the original FOA, the search range of fruit flies will be enlarged, which will greatly enhance the optimization ability of fruit flies.

3.2. Particle Swarm Optimization (PSO). Particle swarm algorithm [28] is a kind of random search algorithm, which is a new intelligent optimization technique, and can converge on the global optimal solution with larger probability. PSO is derived from the study of predatory behavior of birds: a group of birds randomly search for food in a region, all birds know how far they are away from the food, and then the simplest and most effective strategy is to search the surrounding area of birds that is closest to the food. Inspired by this model, it is applied to solve the optimization problem. The basic PSO is as follows:

(1) Suppose in a D-dimensional target search space, N particles form a community, where the i-th particle is expressed as a D-dimensional vector:

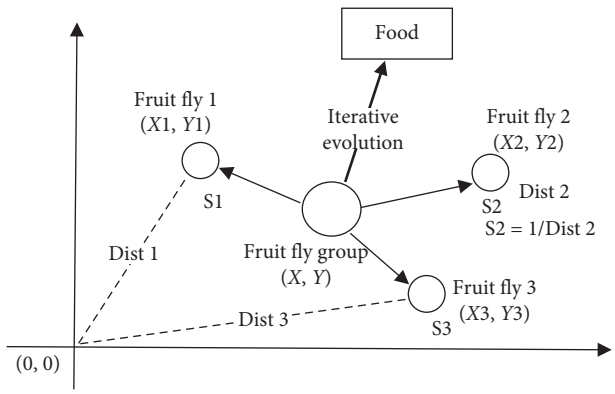


Figure 2: Foraging process of a fruit fly group.

$$Xi = (xi1, xi2, \ldots, xiD), \quad i = 1, 2, \ldots, N. \tag{15}$$

(2) The "flying" velocity of the i-th particle is also a D-dimensional vector, denoted as follows:

$$Vi = (vi1, vi2, \ldots, viD), \quad i = 1, 2, \ldots, N. \tag{16}$$

(3) The optimal position of the i-th particle searched so far is called the individual extremum, denoted as follows:

$$P\text{best} = (pi1, pi2, \ldots, piD), \quad i = 1, 2, \ldots N. \tag{17}$$

(4) The optimal position of the whole particle swarm searched so far is called the global extremum, denoted as follows:

$$g\text{best} = (gi1, gi2, \ldots giD). \tag{18}$$

When these two optimal values are found, the particles will update their speed and position according to the following two formulas:

$$\begin{aligned} Vij(t+1) &= w * vij(t) + c1r1(t)[pij(t) - xij(t)] \\ &\quad + c2r2(t)[pgj(t) - xij(t)], \\ Xij(t+1) &= xij(t) + vij(t+1). \end{aligned} \tag{19}$$

where $c1$ and $c2$ are learning factors, also known as acceleration constants; $r1$ and $r2$ are uniform random numbers within the scope [0, 1], $i = 1, 2, \ldots, D$; vij is the velocity of the particle, $vij \in [-v_{\max}, v_{\max}]$, in which $v_{\max}$ is a constant, and the speed of the particle is set by the user. $r1$ and $r2$ are random numbers between 0 and 1, which increases the randomness of particle flight. W refers to the extent to retain the original speed the greater of the w is, the stronger ability of global convergence and weak ability of local convergence, and the reverse is also true.

The foraging process of a particle swarm group is illustrated in Figure 3 [28].

3.3. Weight Decrease (WD). In this paper, we refer to the weight decrease and random weight algorithm mentioned by Gao [29], and the WD is based on the original PSO and FOA. The larger weighting factor is beneficial to jump out of the local minimum point and is convenient for global search, and the smaller inertia factor is beneficial to the accurate local search of the current search area, which is better for algorithm convergence. Therefore, for the phenomenon that PSO and FOA are easy to get premature and the algorithms are easy to oscillate near the global optimal solution at a later stage, the weight of linear change can be used to reduce the inertia weight linearly from the maximum $\omega_{\max}$ to the minimum $\omega_{\min}$. The formula for the number of iterations with the algorithm is $\omega = \omega_{\max} - (t * (\omega_{\max} \times \omega_{\min}))/t_{\max}$, where $\omega_{\max}$, $\omega_{\min}$, respectively, represent the maximum and minimum values of ω, t indicates the current number of iterations, and $t_{\max}$ indicates the maximum number of iterations.

The weight decrease method can adjust the global and local search capabilities of PSO and FOA, but it still has two shortcomings: first, the local search ability of early iterations is relatively weak, even if the initial particles are close to the global optimal point, it will be missed, and the global search ability will become weak at the later stage, so the program is caught in the local optimal value; second, the maximum number of iterations is difficult to predict, which will affect the adjustment function of the algorithm [30].

3.4. Random Weight (RW). The random weight algorithm is based on the original PSO and FOA. In this paper, the RW refers to taking ω value randomly, so that the impact of the historical speed of particles on the current speed is random. In order to accord with a random number that is randomly distributed ($N(\mu, \hat{\sigma}2)$), the shortcomings of ω linear decrease can be overcome from two aspects. In addition, we can apply the random direction and distance of fruit flies in FOA to increase its global search ability. If the evolution is close to the most power consumption at the beginning of evolution, the linearity of ω decreases, so the algorithm will not converge to the best point, and the random generation of ω can overcome this limitation. ω is calculated as follows:

$$\begin{aligned} \omega &= \mu + \sigma * N(0, 1), \\ \mu &= \mu \min + (\mu \max - \mu \min) * \operatorname{rand}(0, 1), \end{aligned} \tag{20}$$

where N (0, 1) represents the random number of the standard normal distribution, and rand (0, 1) represents a random number between 0 and 1. Researches show that RW-based PSO and FOA algorithm can avoid the local optimum to a certain extent.

4. Case Description

Suppose the length of the shelf is 80 m, the height is 8 m, and a complete shelf has 40 rows and 5 tiers. The lateral movement speed Va of the stacker is 1 m/s and longitudinal velocity Vb is 0.2 m/s. The picking time of each cargo space is assumed to be 10 s. According to the above optimization algorithms, Popsize1 = 5 and Popsize2 = 10, that is, the number of all populations is Popsize1 × Popsize2 = 50. The largest number of iterations of six algorithms is 1000 times. In terms of FOA parameter, the random initial position of a

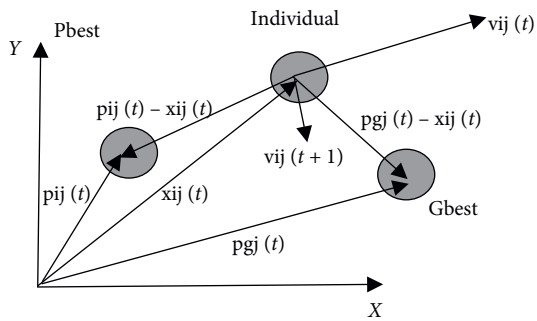


Figure 3: Process of a Particle swarm group.

fruit fly swarm is [−5, 5], fruit flies searching for food randomly, and the distance interval is [−50, 50]; in terms of PSO parameter, $C1$ and $C2$ are set to be 1.49445, $V_{\max}$ and $V_{\min}$ are set to be 1, $\text{pop}_{\max}$ is set to be 50, and $\text{pop}_{\min}$ is set to be −50; six algorithms are run independently of 20 times.

We apply the RW and WD mathematical model to FOA and PSO and take the individual position as the encoding object, and the length of the code is a randomly generated cargo space number. We then assume that the number of subpopulations is Popsize1, the number of individuals in each population is Popsize2, and the number of individuals in all populations is Popsize1 × Popsize2, and then the population quantity is Popsize1 × Popsize2. If m cargo spaces are randomly generated, then the coding scheme of No. b fruit flies in No. a subpopulation is shown in Table 1.

In order to check the optimization capability of the proposed FOA and PSO, two groups of 10 cargo spaces and 20 cargo spaces are randomly generated, as shown in Tables 2–5.

Table 1: Coding scheme of fruit flies randomly generated.

Cargo space	1	2	...	7	8	$m-1$	m
Tier	xab1	xab2	...	xab7	xab8	xab $(n-1)$	xabn
Row	yab1	Yab2	...	yab7	yab8	yab $(n-1)$	yabn
Smell	Sab1	Sab2	...	Sab7	Sab8	Sab $(n-1)$	Sabn

Table 2: 10 cargo spaces in Group 1.

Tier	24	32	40	26	17	12	38	15	7	29
Row	4	1	4	5	3	2	2	3	5	1

Table 3: 10 cargo spaces in Group 2.

Tier	22	32	12	28	40	12	25	34	17	27
Row	2	1	3	3	1	5	3	2	2	4

Table 4: 20 cargo spaces in Group 1.

Tier	20	32	42	35	22	6	19	43	18	38
Row	3	2	3	5	1	3	2	1	4	5
Tier	25	10	44	16	41	17	28	3	7	15
Row	1	5	2	4	4	2	1	3	5	4

Table 5: 20 cargo spaces in Group 2.

Tier	8	20	22	6	12	13	28	14	34	4
Row	4	3	2	4	2	1	3	6	3	1
Tier	33	11	32	3	36	27	40	4	22	25
Row	4	2	2	5	3	1	3	4	2	6

5. Results and Discussion

The results (subfigures) are shown below in proper order: PSO (upper left), WDPSO (center left), RWPSO (lower left), FOA (upper right), WDFOA (center right), and RWFOA (lower right).

5.1. Iteration Verification of 10 Cargo Spaces in Group 1. According to the data of Figure 4, the optimal search time of PSO, WDPSO, and RWPSO is 243 s, 235 s, and 234 s, and the optimal search time of FOA, WDFOA, and RWFOA is 236 s, 228 s, and 226 s.

According to the data of Table 6, the optimal average search time of PSO is 237 s, the optimal search time of FOA is 230 s, and the optimization of FOA is better. The average optimal search time of the original, WD, and RW is 234.5 s, 232 s, and 231 s, respectively, and the optimization of RW is better. Thus, RWFOA is the best.

From the standard deviation in Table 7, RWFOA is the smallest, better than the other five. Therefore, PSO algorithm is featured with good accuracy and speed, but its optimization performance is worse than FOA. For six different algorithms, the optimization of RWFOA is relatively good.

The optimal picking time of 10 cargo spaces is 226 s, and the corresponding picking order is as follows: 8–5–6–9–2–1–10–7–4–3.

5.2. Iteration of 10 Cargo Spaces in Group 2. According to the data of Figure 5, the optimal search time of PSO, WDPSO, and RWPSO is 216 s, 214 s, and 212 s, and the optimal search time of FOA, WDFOA, and RWFOA is 209 s, 208 s, and 207 s.

According to the data of Table 8, the optimal average search time of PSO is 214 s, the optimal search time of FOA is 208 s, and the optimization of FOA is better. The average optimal search time of the original, WD, and RW is 212.5 s, 211 s, and 209.5 s, respectively, and the optimization of RW is better. Thus, RWFOA is the best.

From the standard deviation in Table 9, RWFOA is the smallest, better than the other five. Therefore, PSO algorithm is featured with good accuracy and speed, but its optimization performance is worse than FOA. For six different algorithms, the optimization of RWFOA is relatively good.

The optimal picking time of 10 cargo spaces is 207 s, and the corresponding picking order is as follows: 3–2–1–8–5–7–6–10–9–4.

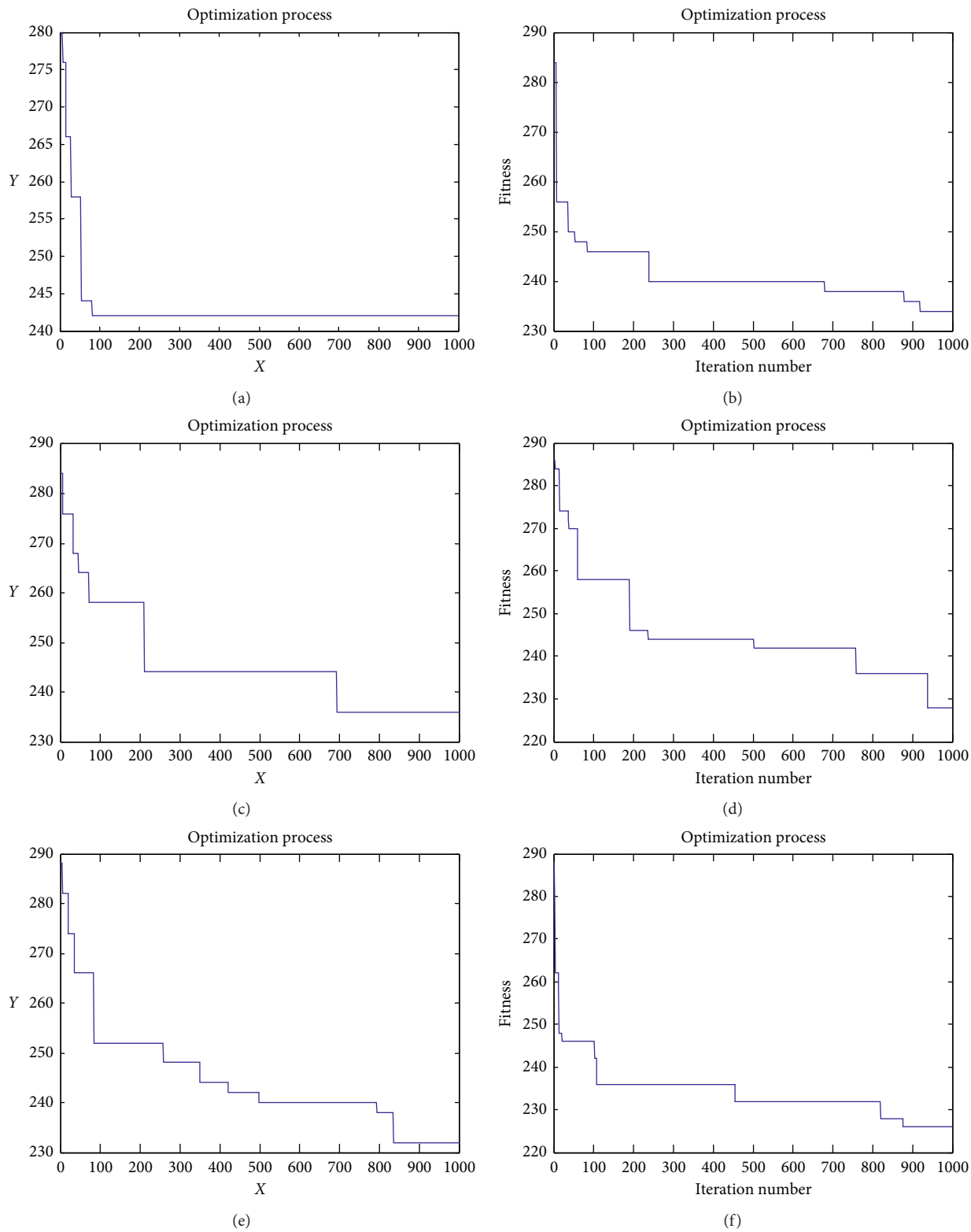


Figure 4: Iterative changes of 10 cargo spaces in Group 1: (a) PSO; (b) FOA; (c) WDPSO; (d) WDFOA; (e) RWPSO; (f) RWFOA. Note: X is the total time of iteration and Y is 1000 iterations.

5.3. Iteration of 20 Cargo Spaces in Group 1. According to the data of Figure 6, the optimal search time of PSO, WDPSO, and RWPSO is 553 s, 550 s, and 549 s, and the optimal search time of FOA, WDFOA, and RWFOA is 545 s, 543 s, and 541 s.

According to the data of Table 10, the optimal average search time of PSO is 550 s, the optimal search time of FOA is 543 s, and the optimization of FOA is better. The average optimal search time of the original, WD, and RW is 549 s,

Table 6: Average of picking time of 10 cargo spaces in Group 1.

Algorithm	Original (s)	WD (s)	RW (s)	Average (s)
PSO	243	235	234	237
FOA	236	228	226	230
Average	234.5	232	231	

Table 7: Standard deviation of picking time of 10 cargo spaces in Group 1.

	Algorithm SD		
Algorithm	Original	WD	RW
PSO	6.6	6.4	6.2
FOA	4.9	3.8	3.7

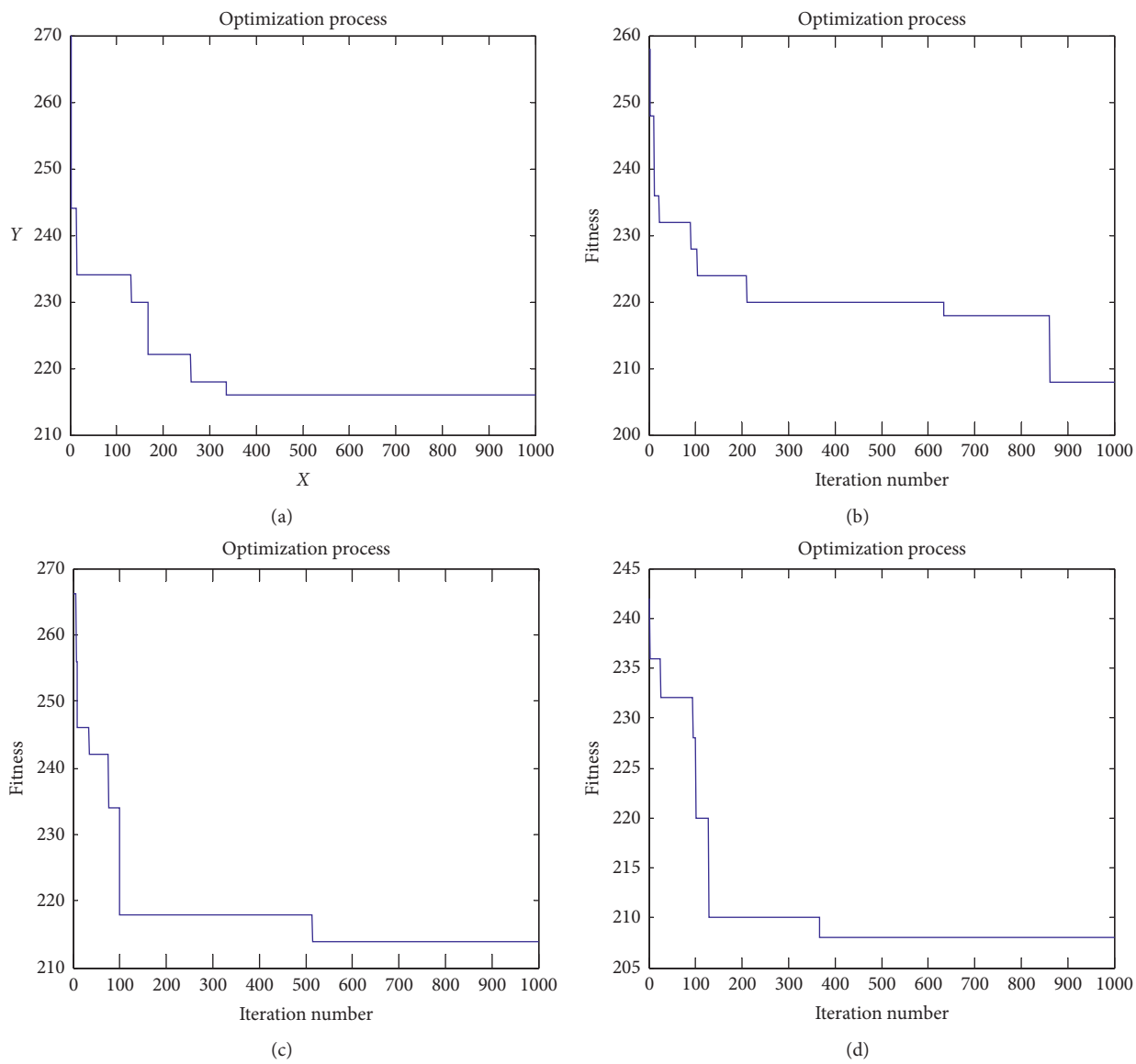
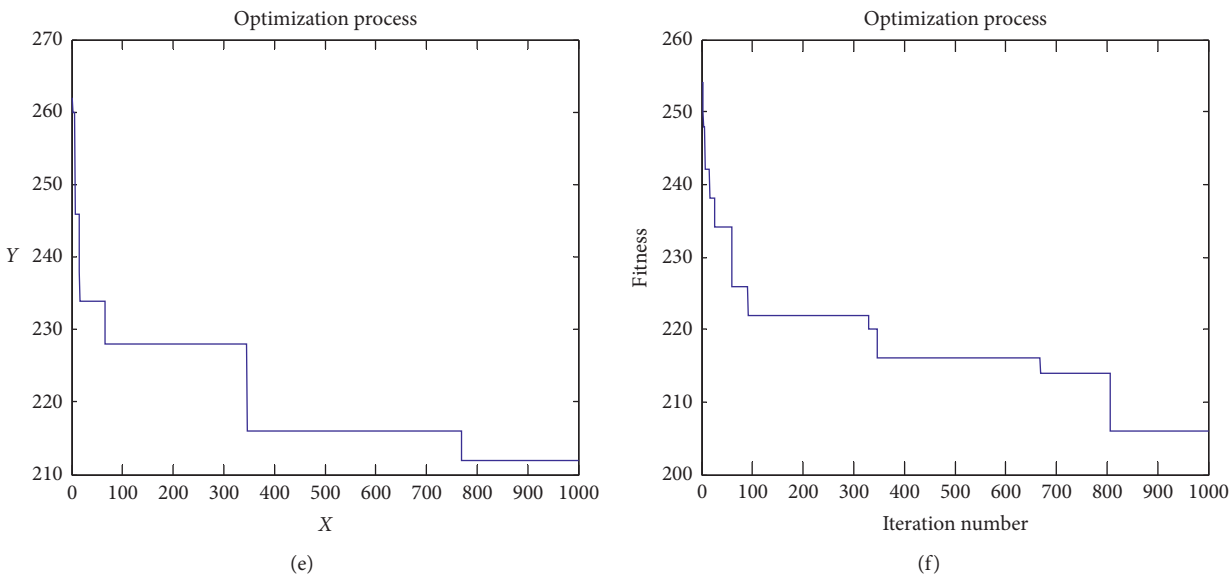


Figure 5: Continued.



(e) (f)

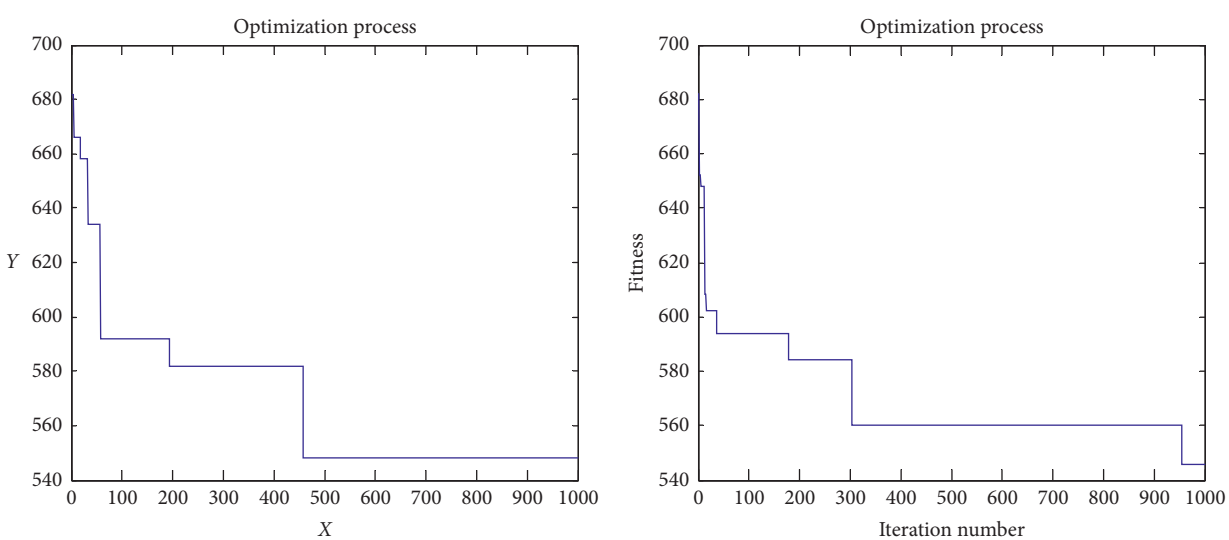
Figure 5: Iterative changes of 10 cargo spaces in Group 2: (a) PSO; (b) FOA; (c) WDPSO; (d) WDFOA; (e) RWPSO; (f) RWFOA. Note: X is the total time of iteration and Y is 1000 iterations.

Table 8: Average of picking time of 10 cargo spaces in Group 2.

Algorithm	Original (s)	WD (s)	RW (s)	Average (s)
PSO	216	214	212	214
FOA	209	208	207	208
Average	212.5	211	209.5	

Table 9: Standard deviation of picking time of 10 Cargo spaces in Group2.

	Algorithm SD		
Algorithm	Original	WD	RW
PSO	6.3	5.9	5.5
FOA	4.8	4.7	4.6



(a) (b)

Figure 6: Continued.

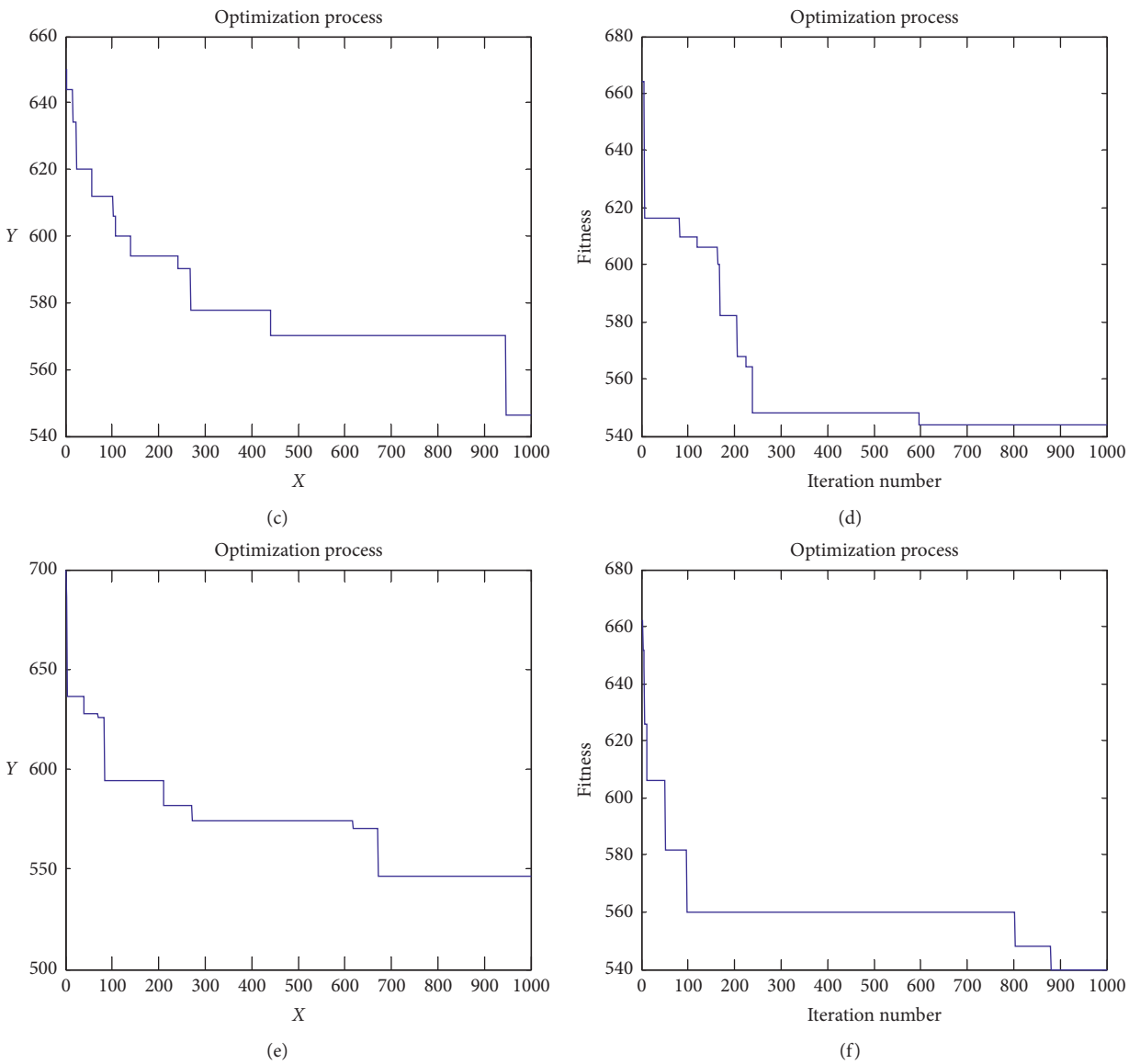


Figure 6: Iterative changes of 20 cargo spaces in Group 1: (a) PSO; (b) FOA; (c) WDPSO; (d) WDFOA; (e) RWPSO; (f) RWFOA. Note: X is the total time of iteration and Y is 1000 iterations.

546.5 s, and 545 s, respectively, and the optimization of RW is better. Thus, RWFOA is the best.

From the standard deviation in Table 11, RWFOA is the smallest, better than the other five. Therefore, PSO algorithm is featured with good accuracy and speed, but its optimization performance is worse than FOA. For six different algorithms, the optimization of RWFOA is relatively good.

The optimal picking time of 20 cargo spaces is 541 s, and the corresponding picking order is as follows: 9–12–19–13–20–15–4–17–8–1–10–2–16–5–14–3.

5.4. Iteration of 20 Cargo Spaces in Group 2. According to the data of Figure 7, the optimal search time of PSO, WDPSO, and RWPSO is 544 s, 542 s, and 540 s, and the optimal search

Table 10: Average of picking time of 20 cargo spaces in Group 1.

Algorithm	Original (s)	WD (s)	RW (s)	Average (s)
PSO	553	550	549	550
FOA	545	543	541	543
Average	549	546.5	545	

Table 11: Standard deviation of picking time of 20 cargo spaces in Group 1.

	Algorithm SD		
Algorithm	Original	WD	RW
PSO	18.4	15.3	15.0
FOA	15.6	14.3	11.2

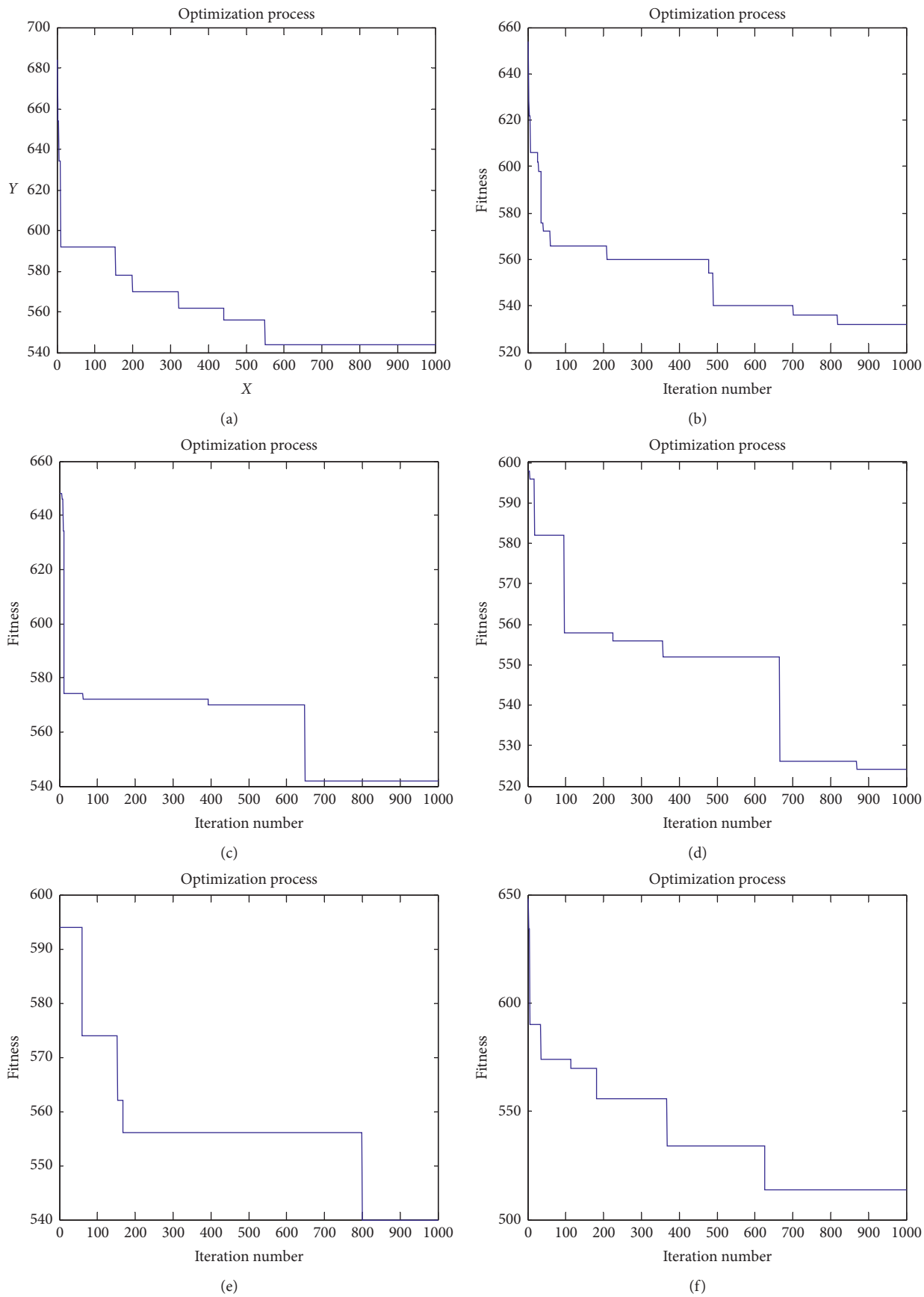


Figure 7: Iterative changes of 20 cargo spaces in Group 2: (a) PSO; (b) FOA; (c) WDPSO; (d) WDFOA; (e) RWPSO; (f) RWFOA. Note: X is the total time of iteration and Y is 1000 iterations.

Table 12: Average of picking time of 20 cargo spaces in Group 2.

Algorithm	Original (s)	WD (s)	RW (s)	Average (s)
PSO	544	542	540	542
FOA	531	528	514	524
Average	537	535	527	

Table 13: Standard deviation of picking time of 20 cargo spaces in Group 2.

	Algorithm SD		
Algorithm	Original	WD	RW
PSO	15.7	14.5	13.2
FOA	21.3	15.5	14.7

time of FOA, WDFOA, and RWFOA is 531 s, 528 s, and 514 s.

According to the data of Table 12, the optimal average search time of PSO is 542 s, the optimal search time of FOA is 524 s, and the optimization of FOA is better. The average optimal search time of the original, WD, and RW is 537 s, 535 s, and 527 s, respectively, and the optimization of RW is better. Thus, RWFOA is the best.

From the standard deviation in Table 13, RWFOA is the smallest, better than the other five. Therefore, PSO algorithm is featured with good accuracy and speed, but its optimization performance is worse than FOA. For six different algorithms, the optimization of RWFOA is relatively good.

The optimal picking time of 20 cargo spaces is 514 s, and the corresponding picking order is as follows: 8–18–19–4–5–1–12–2–10–6–16–15–20–14–11–7–9–3–13–17.

6. Conclusion

With the increasing pursuit of efficiency in logistics warehousing, order picking has also become an important research, and it is constantly proposed to apply a variety of different algorithms to optimize picking time. This paper assumes a model of automated warehouse shelves. By referring to previous studies, the study is designed to set the picking route to get the optimal picking time so as to improve the efficiency of order picking. It has been widely used in various industries, including electronic appliances, pharmaceutical logistics, tobacco logistics, machinery automation, and food industry.

A new FOA, PSO, RW, and WD are used to improve FOA and PSO and to look for the optimal order picking time. The result shows that the optimization capacity of RWFOA is better and the picking efficiency is the best. Therefore, it can be applied to the order picking in automated warehouses, thereby improving warehouse operation efficiency and reducing the time cost of order picking.

RWFOA is a more effective local search method which can be used in future work. The proposed RWFOA could be applied to other variations of the TSP; for example, fixed edges are listed that are required to appear in each solution to the problem, path problem, or vehicle routing problem etc. Therefore, future work could focus on the development of adaptive algorithms with the implementation of other problem-specific features that could improve the performance of the RWFOA.

This study also has certain limitations. For example, the paper assumes that the stacker is moving at a constant speed, but the speed in the actual operating conditions is uncertain. Secondly, this paper takes part of the shelves as the object of study instead of shelf-to-shelf, which means it is the local optimal in the warehouse rather than the global optimal.

Data Availability

The data used to test the algorithm are randomly generated, readers need to pay more attention to intelligent algorithms. Anyway, the data used to support the findings of this study are available from all the authors upon request.

Conflicts of Interest

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

Acknowledgments

This research was financially supported by the 2018 Social Science Planning Project of Guangzhou "Research on the Construction and Development of Guangzhou Smart International Shipping Center Based on the One Belt One Road Strategy" (Grant no. 2018GZGJ169) and 2016 Humanities and Social Sciences Research Projects of Universities in Guangdong Province "Construction of key disciplines in business administration" (Grant no. 2015WTSCX126).

References

[1] R. L. Daniels, J. L. Rummel and R. Schantz, L. J and R. Schantz, "A model for warehouse order picking," *European Journal of Operational Research*, vol. 105, no. 1, pp. 1–17, 1998.

[2] H.-I. Jeong, J. Park, and R. C. Leachman, "A batch splitting method for a job shop scheduling problem in an MRP environment," *International Journal of Production Research*, vol. 37, no. 15, pp. 3583–3598, 1999.

[3] W. Lu, McF. Duncan, V. Giannikas, and Q. Zhang, "An algorithm for dynamic order-picking in warehouse operations," *European Journal of Operational Research*, vol. 248, no. 1, pp. 107–122, 2016.

[4] L. Pansart, N. Catusse, and H. Cambazard, "Exact algorithms for the order picking problem," *Computers & Operations Research*, vol. 100, pp. 117–127, 2018.

[5] J. Bolaños Zuñiga, J. A. Saucedo Martínez, T. E. Salais Fierro, and J. A. Marmolejo Saucedo, "Optimization of the storage location assignment and the picker-routing problem by using mathematical programming," *Applied Sciences*, vol. 10, no. 2, 2020.

[6] H. Hwang, W. J. Baek, and M.-K. Lee, "Clustering algorithms for order picking in an automated storage and retrieval system," *International Journal of Production Research*, vol. 26, no. 2, pp. 189–201, 1988.

[7] M. J. LI, X. B. Chen, and C. Q. Liu, "Solution of order picking optimization problem based on improved ant colony algorithm," *Computer Engineering*, vol. 35, no. 3, pp. 219–221, 2009.

[8] C.-M. Hsu, K.-Y. Chen, and M.-C. Chen, "Batching orders in warehouses by minimizing travel distance with genetic algorithms," *Computers in Industry*, vol. 56, no. 2, pp. 169–178, 2005.

[9] X. Ning and H. Hu, "Multiple-population fruit fly optimization algorithm for scheduling problem of order picking operation in automatic warehouse," *Journal of Lanzhou Jiaotong University*, vol. 33, no. 3, pp. 108–113, 2014.

[10] K. J. Roodbergen and R. Koster, "Routing methods for warehouses with multiple cross aisles," *International Journal of Production Research*, vol. 39, no. 9, pp. 1865–1883, 2001.

[11] D. M.-H. Chiang, C.-P. Lin, and M.-C. Chen, "The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres," *Enterprise Information Systems*, vol. 5, no. 2, pp. 219–234, 2011.

[12] R. De Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: a literature review," *European Journal of Operational Research*, vol. 182, no. 2, pp. 481–501, 2007.

[13] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[14] M. Hahsler and K. Hornik, "TSP infrastructure for the traveling salesperson problem," *Journal of Statistical Software*, vol. 23, no. 1, pp. 1–21, 2007.

[15] C. G. Petersen and G. Aase, "A comparison of picking, storage, and routing policies in manual order picking," *International Journal of Production Economics*, vol. 92, no. 1, pp. 11–19, 2004.

[16] C. Theys, O. Bräysy, W. Dullaert, and B. Raa, "Using a TSP heuristic for routing order pickers in warehouses," *European Journal of Operational Research*, vol. 200, no. 3, pp. 755–763, 2010.

[17] J. Renaud and A. Ruiz, "Improving product location and order picking activities in a distribution centre," *Journal of the Operational Research Society*, vol. 59, no. 12, pp. 1603–1613, 2007.

[18] C. G. Petersen II, "The impact of routing and storage policies on warehouse efficiency," *International Journal of Operations & Production Management*, vol. 19, no. 10, pp. 1053–1064, 1999.

[19] X. Yu, X. Liao, W. Li, X. Liu, and T. Zhang, "Logistics automation control based on machine learning algorithm," *Cluster Computing*, vol. 22, no. 6, pp. 14003–14011, 2019.

[20] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and Q. X. Wang, "Particle swarm optimization-based algorithms for TSP and generalized TSP," *Information Processing Letters*, vol. 103, no. 5, pp. 169–176, 2007.

[21] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[22] Y. Zhang, G. Cui, J. Wu, W.-T. Pan, and Q. He, "A novel multi-scale cooperative mutation Fruit Fly Optimization Algorithm," *Knowledge-Based Systems*, vol. 114, pp. 24–35, 2016.

[23] Y. Zhang, G. Cui, S. Zhao, and J. Tang, "IFOA4WSC: a quick and effective algorithm for QoS-aware servicecomposition," *International Journal of Web and Grid Services*, vol. 12, no. 1, pp. 81–108, 2016.

[24] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[25] H.-z. Li, S. Guo, C.-j. Li, and J.-q. Sun, "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 37, pp. 378–387, 2013.

[26] S.-M. Lin, "Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network," *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 783–791, 2013.

[27] D. Shan, G. Cao, and H. Dong, "LGMS-FOA: an improved fruit fly optimization algorithm for solving optimization problems," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–9, 2013.

[28] T. A. M. Abdel, M. B. Abdelhalim, and S. E.-D. Habib, "Efficient multi-feature pso for fast gray level object-tracking," *Applied Soft Computing*, vol. 14, pp. 317–337, 2014.

[29] F. Gao, *Matlab Intelligent Algorithm Super Learning Manual*, Post & Telecom Press, Beijing, China, 2014.

[30] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the Congress on Evolu-Tionary Computation*, pp. 79–73, Washington DC, USA, July 1998.