IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 6, 2021, accepted August 10, 2021, date of publication August 16, 2021, date of current version September 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105102

A Novel Maximin-Based Multi-Objective Evolutionary Algorithm Using One-by-One Update Scheme for Multi-Robot Scheduling Optimization

SHUJUN YANG[1], YICHUAN ZHANG[1], LIANBO MA[1], YAN SONG[2], PING ZHOU[1], GANG SHI[3], AND HANNING CHEN[4]

[1]Software College, Northeastern University, Shenyang 110016, China
[2]School of Physics, Liaoning University, Shenyang 110136, China
[3]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110169, China
[4]School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

Corresponding authors: Lianbo Ma (malb@swc.neu.edu.cn), Yan Song (profsongyan@163.com), and Hanning Chen (chenhanning@tjpu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1700103; in part by the National Natural Science Foundation of China under Grant 61773103, Grant 61803367, and Grant 61872075; and in part by the Fundamental Research Funds for the Central Universities under Grant N2117005.

ABSTRACT With the continuous development of E-commerce, warehouse logistics is also facing emerging challenges, including more batches of orders and shorter order processing cycles. When more orders need to be processed simultaneously, some existing task scheduling methods may not be able to give a suitable plan, which delays order processing and reduces the efficiency of the warehouse. Therefore, the intelligent warehouse system that uses autonomous robots for automated storage and intelligent order scheduling is becoming mainstream. Based on this concept, we propose a multi-robot cooperative scheduling system in the intelligent warehouse. The aim of the multi-robot cooperative scheduling system of the intelligent storage is to drive many robots in an intelligent warehouse to perform the distributed tasks in an optimal (e.g., time-saving and energy-conserved) way. In this paper, we propose a multi-robot cooperative task scheduling model in the intelligent warehouse. For this model, we design a maximin-based multi-objective algorithm, which uses a one-by-one update scheme to select individuals. In this algorithm, two indicators are devised to discriminate the equivalent individuals with the same maximin fitness value in the environmental selection process. The results on benchmark test suite show that our algorithm is indeed a useful optimizer. Then it is applied to settle the multi-robot scheduling problem in the intelligence warehouse. Simulation experiment results demonstrate the efficiency of the proposed algorithm on the real-world scheduling problem.

INDEX TERMS Many-objective optimization, multi-objective optimization, maximin fitness function, one-by-one update scheme, multi-robot scheduling optimization.

I. INTRODUCTION

Recently, with the widespread application of autonomous robots, these cheap, small and smart robots have been widely employed in the intelligent storage management of the logistic industry [1]. In principle, the basic task of the warehouse system in the intelligent storage management is to transport goods, store goods and distribute goods efficiently [2]. Accordingly, the multi-robot coordination mechanism, if being really efficient and robust, it will make the entire intelligent warehouse management system more efficient [3], [4]. In the multi-robot coordination, a set of individual tasks needs to be scheduled (e.g., allocated and executed) in an optimal way. Such multi-robot scheduling problem can be formulated as a task allocation model where a group of autonomous robots should cater to a set of orders (tasks) in optimal routes satisfied with certain criteria [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Hao Chen.

There are already some heuristic approaches to deal with the multi-robot task scheduling problems for the intelligent storage scenarios [6]. *Milica* have proposed a scheduling problem of a single robot in the process of cargo transportation and employed a whale optimization algorithm (WOA) to resolve [7]. *Elmi* formulated a robotic scheduling problem for a blocked hybrid workshop and solved it by simulated annealing method [8]. A multi-objective evolutionary algorithm is proposed by *Nastasi* to solve a management problem of automated warehouse [9]. *Xing* exploited ant colony optimization algorithm (ACO) to solve the picking path optimization of storage and retrieval machines [10]. *Xin* proposed the problem of multi-robot coordinated path planning in intelligent warehouse [11]. *Li* proposed a multi-task robot scheduling and task allocation optimization problem under the dynamic requirements of customers and employed an improved particle swarm optimization heuristic algorithm (PSO) to solve it [12]. *Chen* proposed an on-line route selection method based on ACO to find picking routes for multiple robots in uncertain picking time [13].

However, most of the existing works focus on minimizing the total time cost of multiple robots without balancing the individual travel cost of each robot [14]–[16]. How to evenly allocate tasks to each robot, reducing the energy consumption of the robots in the warehouse, has become a key issue. For this problem, we need to consider two goals about the robot energy consumption, i.e., to minimize the total energy of all the robots, and to minimize the individual energy of each robot while each robot in the warehouse is not overused.

Therefore, we research the multi-robot cooperative scheduling optimization problem by considering above goals, where the energy consumption of each robot needs to be formulated as an objective function of minimizing the total energy and the individual energy of the robots. Obviously, such multi-robot cooperative scheduling problem is an MaOP [17], where multiple objectives, each of which is specified by a robot, need to be optimized simultaneously. With the increase of the robots, the task allocation of the multi-robot cooperative scheduling model will become increasingly difficult.

Recently, using multi-objective optimization algorithms (MOEAs) to solve real-world MaOPs has become an effective way [18], [19]. Zhou proposed a multi-objective ant algorithm to solve the airline crew rostering problem [20]. Fang used an improved multi-objective evolutionary algorithm for HQPM to solve the multi-objective problem for high quality pattern mining (HQPM), where the objectives are support, occupancy, and utility [21]. Sun proposed the multi-objective immune algorithm to solve the multi-objective scheduling problem [22]. Among those, one effective approach is the indicator-based evolutionary algorithm (IBEA) [23], which employs performance evaluation indicators to ensure the convergence and distribution of the solution set [24], such as HypE [25], R2-MOEA [26], and MOBI-II [27]. Especially, compared with other MOEAs such as Pareto-based algorithms [28]–[30] and decomposition-based algorithms [31]–[33], IBEAs are more effective in solving MaOPs since they will not lose selection pressure in high-dimensional space like Pareto dominance, or need the support of a large number of reference vectors [34]. However, the indicators in IBEAs are rather computationally expensive, especially when dealing with a large number of objectives.

Based on the above consideration, we propose a novel indicator-based algorithm to solve the multi-robot cooperative task scheduling problem, which usually involves more than ten objectives. In this algorithm, the maximin indicator, which is computationally efficient, is used to evaluate the solutions, and a one-by-one update scheme is devised for the environmental selection. Especially, the maximin fitness function has shown some effective properties for the optimization (see Section III).

The main contributions of this paper are as follows.

- A multi-robot scheduling optimization model is proposed in this paper, which is designed based on the energy consumption of the robot, taking into account the consumption of the robots in performing tasks and the additional consumption in the process of task alternation. In particular, for this model, a new chromosome coding method and a special crossover mutation operator are proposed.
- Aiming to solve our proposed multi-robot scheduling optimization model effectively, a novel maximin-based algorithm called MO-MOEA is developed by using one-by-one update scheme and two environmental selection indicators.
- Experiments verify the effectiveness and efficiency of MO-MOEA. Through a large number of comparisons and discussions, it is proved that the performance of our algorithm is more competitive than comparison algorithms.

The rest of this paper is organized as follows. Section II proposes the model of multi-robot cooperative scheduling. In Section III, the maximin fitness function and the proposed algorithm MO-MOEA are shown. Experimental comparisons on benchmark test suites and multi-robot cooperative scheduling optimization problem are provided in Section IV. Section V outlines conclusions and future work.

II. MULTI-ROBOT COOPERATIVE SCHEDULING PROBLEM

A. MODEL

In the intelligent warehouse system, first we pass the task list into the system, and the system completes task assignment through the algorithm, and then the specific storage tasks are handed over to autonomous robots to complete. In principle, these storage tasks can be classified as follows:

(1) The warehousing task, which is the first task type of the system. In this task, goods are transported to empty shelves according to warehouse requirements.

(2) The transferring task, which is the task of moving the goods in the warehouse according to user's demands or warehouse plans.

(3) The shipment task, where goods are moved out of the warehouse according to user demands.

Fig. 1 shows a two-dimensional warehouse environment schematic diagram, where the black grids indicate that goods are placed on the shelves, and white grids indicate that the shelves are empty. The autonomous robot moves along the aisle area to reach each shelf position. In general, the storage area of the warehouse is regarded as a two-dimensional planar grid map. Using this approach, a two-dimensional coordinate system can be constructed, where goods, robots and obstacles can be uniformly marked and calculated. For the convenience of display, all goods are received from the left side of the warehouse and transported out from the bottom of the warehouse. Table 1 shows the list of tasks that the robots need to complete. Based on the schematic diagram of this intelligent storage environment, the multi-robot scheduling model is established in the followings.

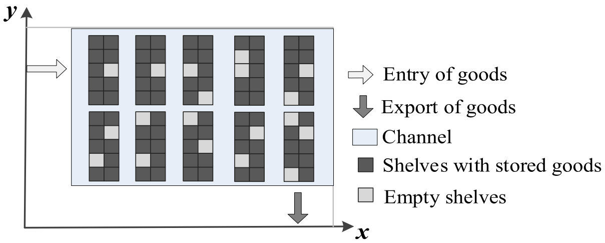


FIGURE 1. Schematic diagram of intelligent warehouse environment.

TABLE 1. Robot's task list.

task sequence	type	location	
1	warehousing	(0,100)	(31,25)
2	transferring	(5,31)	(27,69)
…	…	…	…

Specifically, the multi-robot scheduling model consists of two parts: the task-specific consumption function and the inter-task consumption function. Assume that there are m free moving autonomous robots in an intelligent warehouse $\{r_1, r_2, \ldots, r_m\}$, and n pending tasks to be processed $\{t_1, t_2, \ldots, t_n\}$, which can be assigned to the warehousing tasks, transferring tasks and shipment tasks, respectively. The robot will encounter the following situations when completing its own series of tasks. For the shipment tasks and warehousing tasks, if the coordinates of the task t_i is (x_i, y_i), under the condition without considering the inter-task consumption, the self-cost of autonomous robot performing task t_i can be defined as the function $C_{in}(t_i)$. In the intelligent warehouse environment, the coordinate of the exit is defined as $(f_{out}, 0)$, and the coordinate of the entrance is $(0, f_{in})$. Therefore, the task-specific consumption of the shipment task $C_{out}(t_i)$ is

$$C_{out}(t_i) = 2(|x_i - f_{out}| + |y_i|) \tag{1}$$

The task-specific consumption of the warehousing task $C_{in}(t_i)$ is:

$$C_{in}(t_i) = 2(|x_i| + |y_i - f_{in}|) \tag{2}$$

In addition, if goods are transported from a storage location (x_i, y_i) to another location (x_i', y_i') by an autonomous robot, it is called the transferring task, Different from the above tasks, its task-specific consumption $C_t(t_i)$ is

$$C_{trans}(t_i) = \left|x_i' - x_i\right| + \left|y_i' - y_i\right| \tag{3}$$

In the intelligent warehouse, in addition to the consumption during the execution of the task, there is an additional energy consumption, i.e., the inter-task consumption function. Based on the above model, the Manhattan distance can be used to quickly measure the inter-task consumption of tasks. The inter-task consumption functions are divided into four types: 1) the inter-task consumption between warehousing and shipment tasks; 2) the inter-task consumption between warehousing and transferring tasks; 3) the inter-task consumption between transferring and transferring tasks; 4) the inter-task consumption between shipment and transferring tasks. Since the robot will return to the origin when performing the warehousing task and shipment task, these two can be grouped into one category when calculating the inter-task consumption.

Suppose that an autonomous robot performs a warehousing or a shipment task t_i, the final storage location of the task is located at (x_i, y_i), while the initial storage location of another warehousing or shipment task t_j is (x_j, y_j). Then, the inter-task consumption function $ITC(t_i, t_j)$ of the task interval is

$$ITC(t_i, t_j) = \left|x_j - x_i\right| + \left|y_j - y_i\right| \tag{4}$$

Obviously, for warehousing and shipment tasks, the changes in the order of execution of tasks will not affect its inter-task consumption function, i.e., $ITC(t_i, t_j) = ITC(t_j, t_i)$.

Assume that a robot carries out the transferring task t_m from the storage location (x_m, y_m) to another location (x_m', y_m') while another transferring task t_n from the storage location (x_n, y_n) to another location (x_n', y_n'). Then, the inter-task consumption function $ITC(t_m, t_n)$ between transferring task and transferring task can be formulated as

$$ITC(t_m, t_n) = \left|x_n - x_m'\right| + \left|y_n - y_n'\right| \tag{5}$$

Similarly, the inter-task consumption function $ITC(t_m, t_i)$ between the transferring task t_m and the warehousing task t_i is defined as Eq. (6), and the inter-task consumption function $ITC(t_i, t_m)$ between the warehousing task t_i and the transferring task $\mathrm{t_m}$ is defined as Eq. (7), the inter-task consumption function between the transferring task and the shipment task, is same to the inter-task consumption function between the transferring task and the shipment task:

$$ITC(t_m, t_i) = \left|x_i - x_m'\right| + \left|y_i - y_m'\right| \tag{6}$$
$$ITC(t_i, t_m) = \left|x_m - x_i'\right| + \left|y_m - y_i'\right| \tag{7}$$

For a better efficiency of the task allocation, an efficiency function based on the consumption of the autonomous robots is defined as

$$\begin{aligned} \min f &= (f_1(r_1, S_1), f_2(r_2, S_2), \ldots, f_m(r_m, S_m)) \\ f_i(r_i, S_i) &= \sum_{j=1}^{k-1} ITC(t_{ij}, t_{i(j+1)}) + \sum_{j=1}^{k} C(t_{ij}) \end{aligned} \tag{8}$$

where S_i: $t_{i1} \rightarrow t_{i2} \rightarrow t_{i3} \ldots \rightarrow t_{ik}$ represents the sequence assigned to the robot r_i, which is a linear sequence indicates that the task sequence for the robot r_i consists of k tasks. $f_i(r_i, S_i)$ represents the energy consumption of autonomous robot r_i to complete the given task sequence S_i. After the above design, the final optimization goal is to minimize the total consumption of the robots while minimizing the consumption of each robot.

All consumption functions are evaluated by the Manhattan distance, which can accurately reflect the travel distance of autonomous robots in the warehouse. In this process, the homogeneous autonomous robots with the same speed and energy consumption are employed. The consumption evaluation function based on Manhattan distance not only reflects the energy consumption, but also reflects the task execution time. Therefore, the proposed model covers the two issues in the intelligence warehouse, i.e., the energy consumption and time consumption.

B. CROSSOVER AND MUTATION

In the optimization process of the task sequence, several effective chromosome crossover and mutation operators are designed to obtain better Pareto optimal front.

To effectively reduce the solution space of the task allocation scheme, we use a combined chromosome method, as shown in Fig. 2, which also can enhance the converge performance. In the figure, the task sequences are arranged in order and evenly distributed. The first part of the encoding is the sequence information of the tasks to be executed. The second part is the number of tasks assigned to each autonomous robot, which is composed of multiple randomly generated positive integers, the sum is equal to the total number of tasks to be assigned, and each number represents the number of tasks assigned to each robot. As shown in the figure below, five tasks are assigned to the first robot, then its corresponding task number is the first five task numbers of the first part, and so on.

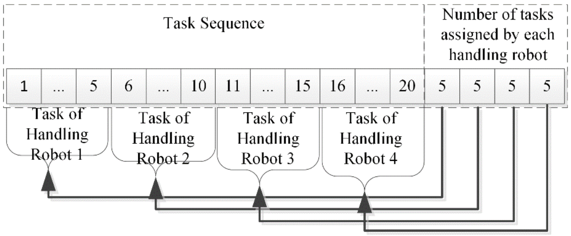


FIGURE 2. Schematic diagram of combinatorial chromosomes.

For the common crossover operators, such as single-point crossover, double-point crossover and multi-point crossover, they cannot guarantee that newly generated individuals are still completely allocated. Therefore, we adopt a combined crossover operator based on the order crossover operator and binary crossover operator. Specifically, the order crossover operator is used for the task sequence part of chromosomes, while the binary crossover operator is for the task number part assigned by each autonomous robot.

The flow of order crossover operator is shown in Fig. 3. First, two genes in a pair of parent chromosomes are selected randomly as the start and end positions, while the selected gene positions of each parent are the same. Then, two offspring are generated, and the genes between the start and end gene positions are consistent with the parents. Finally, the position of the gene selected in the first step is found out in another parent, and then the remaining genes are put in order in the offspring generated in the previous step. This crossover operator can ensure that the newly generated offspring is a scheme of complete task allocation scheme without conflict detection.

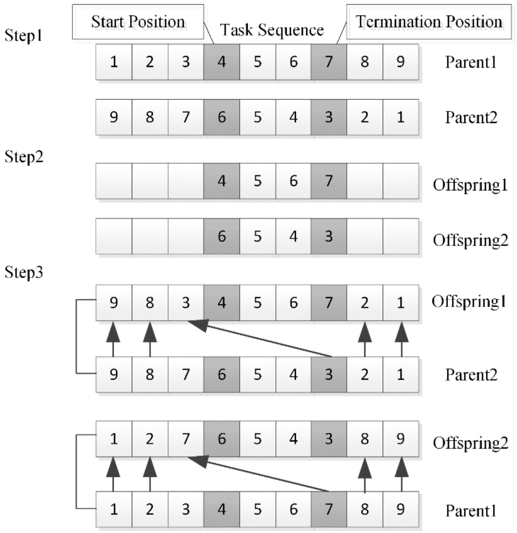


FIGURE 3. Schematic diagram of order crossover.

Unlike the crossover operator, the mutation operator acts on a single gene, and changes the individual effect by changing the value of individual or multiple genes. Generally speaking, the mutation operator can improve the global random search ability of the algorithm, maintain the diversity of the population, and prevent premature convergence. In this paper, the replacement mutation operator is used to mutate the task sequence part of chromosome. The specific process is shown in Fig. 4. Firstly, a sub-bit string is randomly selected from individuals. Then, a mutation site is selected in the remaining bit string. Finally, the selected sub-bit string is inserted into the variant site to form a string of new genes. For the part of number of tasks assigned by each autonomous robot, simple Gaussian mutation can be adopted.

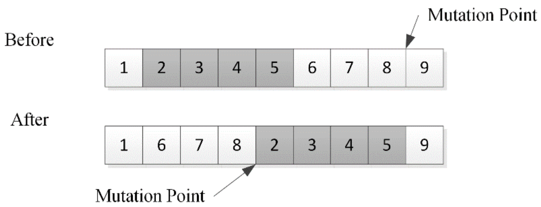


FIGURE 4. Schematic diagram of displacement mutation.

III. PROPOSED METHOD

In the recent literatures, the great potential of IBEAs in solving MaOPs has been demonstrated [35]–[39]. However, the indicators in IBEAs are rather computationally expensive, especially when settling many objectives (i.e., Hypervolume). The maximin fitness function mentioned in this paper is much less computational complexity than Hypervolume. Other advantages and features of the maximin fitness function have been described in the section III.A.

A. MAXIMIN FITNESS FUNCTION

In [40], Balling proposed the maximin fitness function, within which the maximin function is developed to evaluate the individual's fitness. The maximin fitness function is defined as:

$$fitness^i = \max_{j\neq i}(\min_k(f_k(x_i) - f_k(x_j))) \tag{9}$$

where the minimum value is taken from objective 1 to m, and the maximum fitness is taken on all solutions x from population 1 to N, except for the same solution i.

From Eq. (9), we can obtain the following characteristics of the maximin fitness function [40]–[44]:

1) The dominance relation can be determined by the maximin fitness. If the fitness of individual less than or equal to 0 means the individual is non-dominated or weakly-dominated. The fitness of the individual larger than 0 means the individual is dominated.

2) The clusters generated by non-dominated individuals will be penalized by the maximin fitness function, as shown in Fig. 5(a).

3) The maximin fitness of dominated individuals measure their distance to the non-dominated front. See Fig. 5(b).

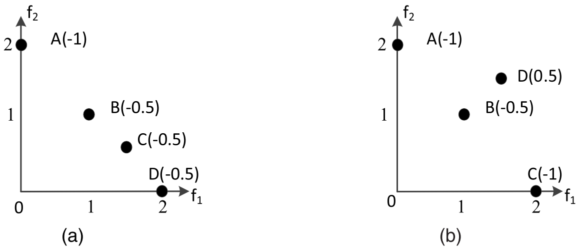


FIGURE 5. Characteristics of the maximin fitness function.

For multi-objective and many-objective optimization, the above characteristics are useful to select promising solutions with good diversity and convergence in the multi-objective search space. However, in Eq. (9), the maximin fitness of the non-dominated individuals are not only controlled by other non-dominated individuals, but also by the dominated individuals, as shown in Fig. 5(b), which may lower the performance of search Pareto-optimal solutions.

Then, we can obtain a modified maximin fitness function [43]:

$$fitness^i = \max_{j\neq i, j\in ND}(\min_k(f_k(x_i) - f_k(x_j))) \tag{10}$$

where *ND* is the non-dominated solution set. When Eq. (10) is employed to evaluate individuals in the population, we can see each non-dominated individual is only affected by other non-dominated individuals. For example, in Fig. 5(b), individual B is no longer controlled by individual D, so the maximin fitness of individual B equals to -1.

B. ONE-BY-ONE UPDATE SCHEME

Through the maximin function, the contribution of individuals to the convergence of the population, as well as the distribution of individuals in the population can be reflected. However, if the indicator is used directly to evaluate the quality of individuals in the population, the process of individual selection will face the dilemma of equivalent selection. That is, there may be individuals with the same fitness value and cannot be selected. As shown in Fig. 6, individuals A to E are evenly distributed in the population, so they have the same fitness. However, in the environmental selection process, not all individuals will be retained. For example, keep three of them. Because all individuals have the same convergence, it is necessary to consider the distribution, individuals A, C, and individual E should be retained, and individuals B and D should be discarded. However, if only considering maximin fitness, the algorithm may not be able to make the correct selection operation, which will affect the optimization efficiency of the entire algorithm. To this end, this section will propose a one-by-one update scheme.

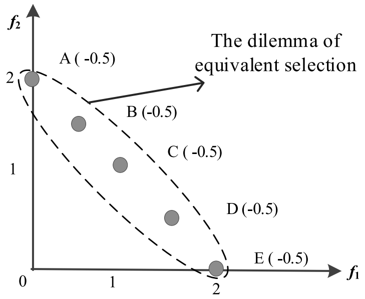


FIGURE 6. Schematic diagram of the dilemma of equivalent selection.

By using the one-by-one update scheme, we can effectively deal with the dilemma of equivalence selection. The strategy steps are as follows. First, use the maximin fitness function to select the first individual of the new population, the maximin fitness function is used to calculate the individuals' maximin fitness according to the Eq. (11). Then the one-by-one update scheme is adopted to select the individual with the least fitness among the candidate individuals to the new population. In this process, the new population can maintain good distribution and convergence. As shown in Fig. 7, individual0073 A to E are uniformly arranged in the two-dimensional objective space, then three individuals will be uniformly selected from them using the one-by-one update scheme. In the first selection, since there is no selected individual, set the ideal point as the reference point to calculate the individuals' maximin fitness according to Eq. (11), as shown in Fig. 7(a). According to the fitness, individual A or individual E is selected, and we assume individual A is selected. In the second selection, according to the selected individual A, the fitness of the remaining individuals will be calculated. As shown in Fig. 7(b), since the fitness of individual E is the smallest, individual E is selected. In the third selection, the selected population includes individual A and individual E, while the candidate individuals include individual B, individual C, and individual D. After that, calculate the fitness of the candidate individuals, the result is shown in Fig. 7(c), individual C has the smallest fitness and is selected. Finally, through this scheme, individual A, individual C, and individual E are selected, as shown in Fig. 7(d), maintaining a good distribution.

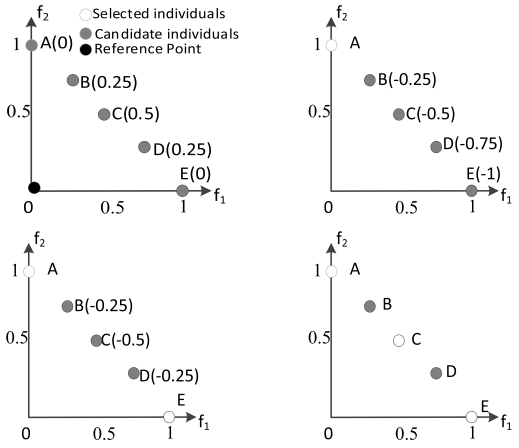


FIGURE 7. Schematic diagram of environment selection based on one-by-one update scheme.

The following is the formula for selecting the first individual for the new population:

$$fitness\_first^i = \max(\min_k(f_k(x_i) - z_k^*)) \tag{11}$$

where $z*$ is the ideal point after normalization.

C. PROPOSED ALGORITHM:MO-MOEA

In this section, the overall framework of the algorithm is given. The specific execution steps of Algorithm 1 are as follows.

First, N individuals are randomly initialized as the initial parent population (Line 1). Then, based on the maximin fitness value of each individual, a sufficient number of individuals are added to the mating pool (Line 3). Then, the simulated binary crossover operator and polynomial mutation operator are used to perform crossover and mutation operation to generate offspring population P'' (Line 4). After that, a certain number of individuals are selected from the combined population Q through the *Environmental Selection* process to form a new population P (Line 6). Continue the above process until the given termination condition is reached. In the following sections, the environmental selection process will be introduced in *Algorithm* 2.

Algorithm 1 General Framework of MO-MOEA

```
1: Objective normalization ($P$)
2: while the given termination condition is not reached do
3: $P' = \text{MatingSelection}\ (P, N)$;
4: $P'' = \text{Variation}\ (P', N)$;
5: $Q = P \cup P''$
6: $P = \text{Environmental\_Selection}\ (Q, N)$;
7: end
8: return $P$
```

1) NORMALIZATION

In this paper, the *Normalization* is also indispensable, which can be used to improve the algorithm's ability to solve scaled problems. Normalization has proved to be helpful in solving the problem of different value ranges for each objective [45]. In the normalization process, the formula is defined as

$$f'_m(x) = \frac{f_m(x) - z_m^*}{z_m^{nad} - z_m^*} \tag{12}$$

where z_m^* is the ideal objective, z_m^{nad} is the nadir objective, m represents the index of the objective dimension, $m \in \{1, 2, \ldots, M\}$. Due to the Pareto optimal front is unknown, we estimate the ideal objective and nadir objective from the solution set.

2) ENVIRONMENTAL SELECTION

The steps of the *Environment Selection* are shown in *Algorithm* 2. First, objective normalization is implemented for each objective, the formula is defined in Eq. (12) (Line 1 in *Algorithm* 2). Then, the individuals are selected according to the similarity verifying method (Line 2 in *Algorithm* 2). Then the number of non-dominated individuals ND in the population S is determined according to the characteristic 1 of the maximin fitness function (Line 3 in *Algorithm* 2). If the size of non-dominated individuals is larger than N, we set the ideal point as the reference point. According to Eq. (11), the individuals with the smallest fitness are added to the new population P. Then, by calculating the maximin fitness between the new population P and the candidate individuals in the population S according to the Eq. (9), the individuals are selected one by one to the new population P (Line 5-15 in *Algorithm* 2). If the number of non-dominated individuals is not larger than the required number N, first, non-dominated

Algorithm 2 Environmental_Selection (Q, N)

```
Input: $Q$ (combined population), $N$ (population size)
Output: $P$ (output population)
1. Objective normalization ($S$);
2. $S$ = similarity_verify ($S$, min_dif)
3. $ND = \text{Sort}(S)$; /∗ The number of the non-dominated solutions are determined according to the Eq. (9) ∗/
4. if $|ND| > N$
5. $z^*$: the ideal point of non-dominated solutions
6. $P^1 = \arg\min_{\mathrm{i}=\{1,2,\ldots,|S|\}}\{fitness\_first^i\}$;
7. $S = S\ P^1$;
8.   for i = 1: N-1
9.     $MF = \arg\min_{\mathrm{i}=\{1,2,\ldots,|S|\}}\{fitness^i\}$;
10.    if $|MF| > 1$
11.      $MS$ = Niching ($MF$, $P$);
12.      $P = P \leq MS$;
13.    else
14.      $P = P \leq MF$;
15.    end if
16.  end for
17. else
18. $P = P \cup ND$;
19. $US = S\ P$;
20.  for i = 1: N-$|ND|$
21.    $MF = \arg\min_{\mathrm{i}=\{1,2,\ldots,|US|\}}\{fitness^i\}$;
22.    if $|MF| > 1$
23.      $MS$ = Niching ($MF$, $P$);
24.      $P = P \leq MS$;
25.    else
26.    $P = P \leq MF$;
27.    end if
28.  end for
29. end if
30. end for
31. return$P$
```

Algorithm 3 Niching

```
Input: $MF$: sloutions which have same maximin fitness
Output: $MS$: solution with minimal angle and Euclidean distance
1. $ES = \arg\min_{x_i \in MF} DIS(x_i)$;/∗ ES is a set of solutions with same Euclidean distance to the ideal point ∗/
2. if $|ES| > 1$
3.   $MS = \text{argmax}_{x_i \in ES, y \in P}\text{angle}(x_i, y)$;
end
5. if $|MS| > 1$
6.   $MS$ = random ($MS$);
7. end
8. return$MS$
```

individuals are added to the new population P. After that, according to the maximin fitness of the candidate individuals, individuals are one by one selected to the new population P (Line 18-26 in *Algorithm* 2). The procedure ends when the number of individuals in the new population exceeds the population size. And *Niching* (Line 11,23 in *Algorithm* 2) is designed to further address the equivalent selection problem, the detail is described in *Algorithm* 3.

3) TWO PERFORMANCE INDICATORS

Although one-by-one update scheme can be used to select individuals that are more suitable for the new population, it may still have a small probability of facing the equivalent selection problem, so other methods are needed to select better individuals from individuals which have same maximin fitness. The following two performance indicators are employed to further select individuals. First of all, convergence is an essential evaluation indicator when solving MaOPs. The Euclidean distance from the candidate individual to the ideal point obtained by Eq. (13) is applied to evaluate the convergence of the candidate individual. This indicator can effectively evaluate individuals with better convergence in the objective space [46].

$$DIS(x) = \sqrt{\sum\nolimits_{m=1}^{M} f_m^2(x)} \tag{13}$$

The individual with the smallest *DIS* value is regarded as the closest one to the ideal point after normalization, and it will be given priority in the next selection.

If the candidate individuals have the same *DIS* value at this time, considering from the perspective of population diversity and propose a diversity indicator to evaluate individuals according to Eq. (14). In high dimensional space, cosine value is effective to measure the distribution of the individuals. In this paper, the angle of the individual in the objective space is used to measure the diversity, the formula is as follows:

$$angle(x, y) = \arccos(\tfrac{f(x)^T f(y)}{||f(x_i|||||f(y)||}) \tag{14}$$

where $f(x)$ and $f(y)$ are the normalized vectors of the two individuals x and y in the objective space, respectively.

Niching is used to further solve the dilemma of equivalent selection, as shown *Algorithm* 3. In this part, we use two indicators to evaluate the individuals with the same maximin fitness, called the same equivalent individual set (*ES*). First, the Euclidean distance to the ideal point of the *ES* were compared, the individual with minimal distance to the ideal point will be selected, called individuals with minimal Euclidean distance (*MS*). If the size of *MS* is larger than 1, the acute angle between *MS* and population *P* is used to measure the distribution of *MS* to the population *P*. Only one individual will be selected and added to the new population *P*. If more than one individual remains after selection by these two indicators, an individual is randomly selected and added into the new population *P*.

4) SIMILARITY VERIFICATION

In *Algorithm* 4, we proposed an approach to verify the individuals' similarity. The individuals in *S* are selected one by one if their similarity is smaller than *min_dif*, which is defined in [43] (Lines 2-9 in *Algorithm 4*).

Algorithm 4 Similarity_verification (*S, min_dif*)

```
Input: P: population, min_dif:
Output: S': output population
$S' = \emptyset$, N= $|S|/2$
for i = 1 to $|S|$ do
    for j = 1 to M do /*M is the number of problem's variables*/
        if $|S[i].f_j - S[i].f_j| > min\_dif$
            if $|S'| < NS' = S' \cup S[i]$;
            end
        end
    end
end
return S'
```

D. COMPUTATIONAL COMPLEXITY

First, the time complexity of objective normalization (Line 1 of *Algorithm* 2) is O(mN). The complexity of the similarity verification process (*Algorithm* 4) is O($Nlog2N$) in the number of comparisons. The maximin fitness calculation of the population requires *O(mN)* additions. The time complexity of using the maximin function to one by one select individuals is O(mN^2). In the worst case, the computational complexity of the niching process of *Algorithm* 3 is O(N_{ES}), where N_{ES} is the amount of *ES*. And N_{ES} $\ll$N. And the *Algorithm* 3 is rarely used. In a word, the overall computational complexity of one generation in MO-MOEA is $O(mN^2)$.

IV. EXPERIMENTAL STUDY

This section includes two parts: 1) preformation validation on benchmarks, which exhibits the advantages of the proposed algorithm in many-objective optimization; 2) application on multi-robot cooperative scheduling optimization problem [47], [48].

A. EXPERIMENTAL RESULTS ON BENCHMARKS

This section gives comparison results between MO-MOEA and other five start-of-art MOEAs, i.e., MOEA/D [31], MOMBI-II [27], NSGA-III [49], RVEA∗[50], and MD-MOEA [43], on the benchmark test suite MaF [51]. These contrast algorithms include three mainstream methods: 1) Pareto-based algorithm, i.e., NSGA-III, 2) decomposition-based algorithm, i.e., MOEA/D, RVEA∗, and 3) indicator-based algorithm i.e., MOMBI-II and MD-MOEA. We use these classical algorithms to compare with our proposed algorithm to verify its effect. Experiments are performed on PlatEMO with MATLAB R2018b [52,53]. Each test problem is tested in 5-, 8-, 10- and 15-objective instances. Then, each algorithm is run for 20 times on each test instance independently [54]. We verify the performance of the algorithm from two aspects: the image obtained by the algorithm solving the instance and the widely used performance indicators.

1) BENCHMARK TEST PROBLEMS AND PERFORMANCE INDICATORS

MaF is a widely used test suite, which is adopted for empirical comparisons in this paper. And MaF test suite includes some ''convex'' DTLZ problems [55] and several WFG problems [56]. Compared with other test suites, MaF test suite shows some characteristics that are more in line with actual problems, which represent the real scene well, and can more effectively verify the effectiveness and effect of MO-MOEA. This test suite is composed of optimization problems with many different Pareto front. Therefore, the MaF test suite is used for comparison between algorithms.

In this paper, we conduct experiment on problems with 5, 8, 10, and 15 objectives. The number of decision variables is set to M + 9 for MaF1-6, MaF10-12, M + 19 for MaF7, 5 for the MaF13 and 2 for MaF8, 9. For the large-scale case MaF14,15, the number of decision variables are M∗20. A detailed description of the parameters can be found in [51].

In this work, we use IGD as metric to evaluate the experimental results of the algorithms. IGD is a comprehensive indicator to measure the convergence and distribution of the solution sets. For more detail of metric IGD, refer to [57], [58].

2) EXPERIMENTAL CONFIGURATION

1) Population size: As we all know, the population size in MOEA/D, MOMBI-II, RVEA∗ and NSGA-III cannot be arbitrarily defined. They are all specified by a two-level reference point strategy. For the fairness of comparison, the population size of other algorithms to the same value as the previous algorithms. See the Table 2.

TABLE 2. Population size of different objective numbers.

m	H1	H2	N
5	5	0	210
8	3	2	156
10	3	2	275
15	2	1	135

2) Crossover and mutation: We use the commonly used simulated binary crossover (SBX) [59] and polynomial mutation [60] for crossover mutation. The probability of crossover and mutation is set to 1.0 and 1/D (where D is the number of decision variables). Set the distribution index of SBX and polynomial mutation to 20 [61].

3) Termination Conditions: Each algorithm runs independently 20 times. For MaF1-15, the maximum number of fitness evaluations (*FEs*) of the algorithm is determined by the decision variable, which is max {100000, 10000 × D}. Details can refer to [49].

4) Other parameters: For MOEA/D, the Tchebycheff approach is used and the neighborhood range is set to N/10. On the algorithms MD-MOEA and MO-MOEA, $min\_dif = 0.0001$ (minimum difference) are employed.

TABLE 3. The mean and standard deviation of IGD on several MaFs.

Problem	objs	MO-MOEA	MOEA/D	NSGA-III	MOMBI-II	MD-MOEA	RVEAa
MaF1	5	1.0760e-1 (2.18e-4)	1.2604e-1 (2.12e-4)-	1.8252e-1 (9.40e-3)-	2.2811e-1 (1.39e-4)-	1.1492e-1 (1.76e-3)-	1.4732e-1 (8.23e-3)-
	10	2.5591e-1 (9.88e-4)	5.0250e-1 (2.57e-2)-	2.8474e-1 (5.24e-3)-	3.7453e-1 (1.01e-2)-	2.4459e-1 (4.65e-3)+	3.2433e-1 (2.55e-2)-
MaF2	5	9.1114e-2 (1.58e-3)	1.1122e-1 (1.29e-4)-	1.1287e-1 (3.72e-3)-	1.5669e-1 (1.33e-3)-	8.9674e-2 (2.00e-3)+	8.3173e-2 (1.15e-3)+
	10	1.6536e-1 (3.63e-3)	2.6171e-1 (3.36e-4)-	2.2101e-1 (2.14e-2)-	6.0170e-1 (2.14e-1)-	1.6623e-1 (3.30e-3)-	2.1961e-1 (7.42e-3)-
MaF5	5	1.8677e+0 (3.53e-2)	7.1241e+0 (2.01e+0)-	1.9701e+0 (1.58e-3)-	1.9491e+0 (8.36e-3)-	1.7118e+0 (1.77e-2)+	2.0817e+0 (3.24e-1)-
	10	4.6913e+1 (1.81e+0)	3.0201e+2 (1.32e+0)-	7.8642e+1 (4.65e-1)-	8.5945e+1 (2.81e+0)-	5.8337e+1 (3.61e+0)-	8.6076e+1 (6.59e+0)-
MaF6	5	4.4297e-3 (2.21e-4)	5.7226e-2 (1.35e-1)-	1.8329e-2 (3.45e-3)-	1.5782e-1 (3.98e-2)-	3.4382e-3 (2.48e-4)+	2.1568e-2 (9.06e-3)-
	10	6.6909e-1 (2.23e-1)	1.0739e-1 (2.22e-1)+	3.0940e-1 (5.74e-2)+	6.7790e-1 (1.10e-1)-	3.1504e+1 (1.32e+1)-	2.7599e-2 (8.07e-3)+
MaF7	5	2.4885e-1 (4.39e-2)	5.4417e-1 (1.09e-2)-	2.8255e-1 (9.56e-3)-	3.9967e-1 (7.28e-2)-	2.6813e-1 (1.08e-2)-	2.1528e-1 (4.26e-3)+
	10	7.9420e-1 (5.37e-2)	3.2344e+0 (4.83e-1)-	9.7765e-1 (5.05e-2)-	4.5968e+0 (9.62e-1)-	1.0077e+0 (1.62e-2)-	8.5375e-1 (6.51e-2)-
MaF8	5	7.4327e-2 (7.65e-4)	1.9014e-1 (2.42e-2)-	1.6458e-1 (1.00e-2)-	3.2880e-1 (8.34e-3)-	9.6054e-2 (3.49e-3)-	2.7299e-1 (6.93e-2)-
	10	9.9067e-2 (7.04e-4)	9.4096e-1 (2.37e-2)-	2.9202e-1 (6.78e-2)-	1.4633e+0 (4.63e-2)-	1.1676e-1 (2.13e-3)-	9.2053e-1 (2.38e-1)-
MaF9	5	7.1743e-2 (6.43e-4)	9.9880e-2 (4.48e-3)-	4.9753e-1 (1.87e-1)-	3.8438e-1 (4.89e-2)-	1.0534e-1 (4.61e-3)-	1.9229e-1 (3.99e-2)-
	10	9.9479e-2 (4.54e-4)	1.1101e+0 (1.49e+0)-	5.2065e-1 (1.26e-1)-	1.4834e+0 (4.55e-2)-	1.4778e+1 (1.06e+1)-	8.3080e-1 (1.83e-1)-
MaF10	5	3.6958e-1 (5.98e-3)	7.5834e-1 (3.00e-2)-	3.9296e-1 (1.28e-2)-	4.0955e-1 (8.71e-3)-	5.7603e-1 (3.07e-2)-	4.9495e-1 (3.92e-2)-
	10	9.0965e-1 (1.14e-2)	1.7805e+0 (9.52e-2)-	1.0457e+0 (5.43e-2)-	1.2036e+0 (3.33e-2)-	2.5588e+0 (2.57e-1)-	1.0094e+0 (4.23e-2)-

TABLE 4. Average robot consumption provided by the algorithm in the 10-objective robot scheduling problem.

	MO-MOEA	MD-MOEA	IBEA
Robot 1	1.8995E+04	2.5717E+04	2.6584E+04
Robot 2	2.1633E+04	2.1658E+04	1.7958E+04
Robot 3	2.5189E+04	1.9482E+04	2.3258E+04
Robot 4	2.4696E+04	2.1491E+04	2.0585E+04
Robot 5	1.6207E+04	2.0662E+04	1.8876E+04
Robot 6	1.8519E+04	2.1363E+04	1.8798E+04
Robot 7	1.9220E+04	1.8829E+04	2.3317E+04
Robot 8	2.1250E+04	2.0729E+04	2.6428E+04
Robot 9	2.6895E+04	2.3715E+04	1.6328E+04
Robot 10	1.8876E+04	1.9746E+04	2.0756E+04
Total	2.1148E+05	2.1339E+05	2.1288E+05

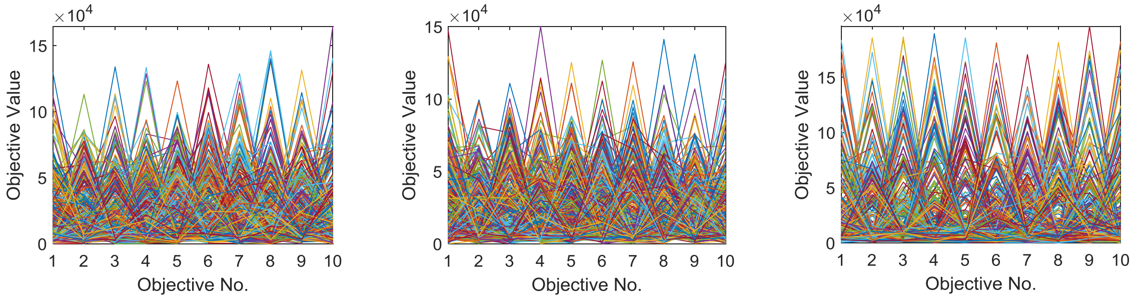
3) EXPERIMENTS AND ANALYSIS ON MAFS

In this section, the performance of MO-MOEA in solving many-objective optimization problems is verified on the MaF benchmark problems. Table 3 shows the experimental results of the six algorithms after 20 runs and highlights the best results (completed results are provided in the supplementary file). The Wilcoxon rank sum test is used, where the significance level is set to 0.05, and the statistics are performed. The symbols "+", "−", and "≈" respectively indicate that the compared MOEA performs significantly better, significant worse or significantly similarly to the MO-MOEA.

As shown, we can see that MO-MOEA has demonstrated strong competitiveness, achieving first ranks on most of the MaF instances. MD-MOEA performs poorly on the MaF test suite. The reason may be that MD-MOEA uses a one-time comparison method to select the solution. By contrast, the reason for the promising performance of MO-MOEA is that one-by-one update scheme makes a better diversity between non-dominated individuals. This scheme ensures that the algorithm considers convergence and diversity in the process of selecting individuals. Through the above experiments, it can be determined that MO-MOEA has performed

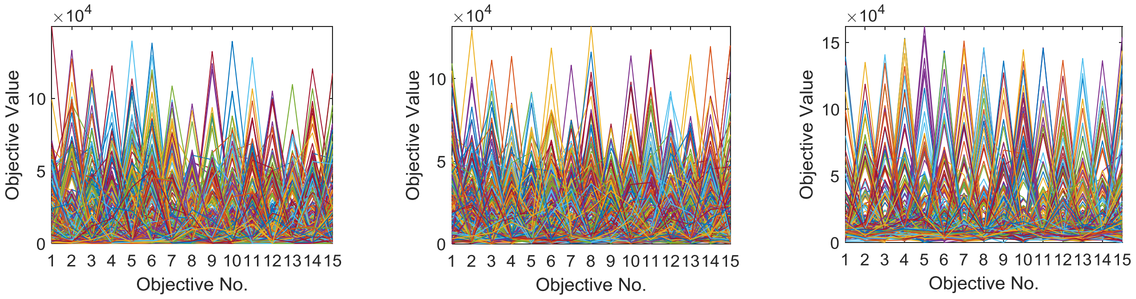
TABLE 5. Average robot consumption provided by the algorithm in the 15-objective robot scheduling problem.

	MO-MOEA	MD-MOEA	IBEA
Robot 1	1.3854E+04	1.9675E+04	1.2608E+04
Robot 2	1.6118E+04	1.3849E+04	1.4085E+04
Robot 3	1.3871E+04	1.5346E+04	1.4132E+04
Robot 4	1.3005E+04	1.3219E+04	1.6671E+04
Robot 5	1.5768E+04	1.3128E+04	1.5774E+04
Robot 6	1.8038E+04	1.4096E+04	1.3284E+04
Robot 7	1.2583E+04	1.4556E+04	1.3876E+04
Robot 8	1.2756E+04	1.3155E+04	1.3170E+04
Robot 9	1.3045E+04	1.2744E+04	1.4693E+04
Robot 10	1.4739E+04	1.3181E+04	1.5154E+04
Robot 11	1.4669E+04	1.2423E+04	1.2769E+04
Robot 12	1.2797E+04	1.4483E+04	1.3935E+04
Robot 13	1.2505E+04	1.3924E+04	1.2098E+04
Robot 14	1.3489E+04	1.5243E+04	1.3567E+04
Robot 15	1.5210E+04	1.3522E+04	1.7021E+04
Total	2.1247E+05	2.1254E+05	2.1283E+05



(a) MO-MOEA (b) MD-MOEA (c) IBEA

FIGURE 8. Final solution set of involved algorithms on the 10-objective robot scheduling problem, shown by parallel coordinates.



(a) MO-MOEA (b) MD-MOEA (c) IBEA

FIGURE 9. Final solution set of involved algorithms on the 15-objective robot scheduling problem, shown by parallel coordinates.

well in solving MaOPs, and MO-MOEA has also shown high efficiency on large-scale problems.

4) OTHER EXPERIMENTAL RESULTS

The parallel coordinates of the final solutions of the algorithm on 15 objectives on several MaFs are provided in the supplementary file. As shown, the compromise surface formed by the solution finally generated by MO-MOEA is better than the surface generated by other algorithms in both convergence and distribution, while MD-MOEA and RVEA∗

are also competitive, but the effect is not very good from the perspective of distribution.

B. EXPERIMENT RESULTS ON MULTI-ROBOT SCHEDULING PROBLEM

The benchmark test suite proves that MO-MOEA is effective in many-objective optimization. Then, the proposed algorithm is used to optimize the multi-robot scheduling problem of intelligent warehouse. Suppose that in the intelligent warehouse with a length and width of 100m, the goods entrance of the warehouse is at (0, 100) and the exit of the warehouse is at (100, 0). There are 10 or 15 autonomous robots in the intelligent warehouse, all of which run at a speed of 1m/s and are initially located near the origin of coordinates. There are 1,000 storage tasks to be processed, including 300 warehousing tasks, 300 shipment tasks, and 400 transferring tasks in the intelligence warehouse. All tasks are generated randomly.

To ensure that the experimental results are not affected by randomness, the task list is executed 10 times independently by the above algorithm MO-MOEA and the other two indicator-based algorithms IBEA and MD-MOEA. The population size of these algorithms is 500, and the maximum fitness evaluations is 100000.

Table 4 and Table 5 show the average motion distance of each robot optimized by MO-MOEA, IBEA and MD-MOEA. They also represent the energy consumption of each robot to perform storage tasks. The above algorithms all realize the scheduling and allocation of all storage tasks. It can be found from the data that the consumption function values of all autonomous robots are on the same order of magnitude, so the load of each robot is basically balanced.

Fig. 8 and Fig. 9 show the coordinates of the final solutions formed by the algorithms in the 10− and 15-objective multi-robot scheduling optimization problem. As can be seen from the figures that the compromise surface formed by the solutions finally generated by MO-MOEA is superior to the compromise surface generated by other algorithms in terms of convergence and distribution, the energy consumption of robots obtained by MO-MOEA is smaller in general. It can be seen from the figure that the solution set optimized by the MO-MOEA algorithm rarely has extreme solutions, that is, it is rare that one robot undertakes most of the tasks and other robots undertake a small part of the tasks.

We employ HV [62] to determine the effect of the algorithm on actual problems. Choose a set of maximum values as the reference point for each objective of HV, such as 200000 for each objective. Generally, the HV value of MO-MOEA is larger than traditional indicator-based algorithm IBEA and maximin-based algorithm MD-MOEA. In other words, MO-MOEA can obtain a solution set with better distribution and convergence. Through such experiments, it can be learned that MO-MOEA can provide managers with more diverse choices from the similar efficient plans. After the above experiments, it can be shown that the proposed algorithm has also achieved good results in solving multi-robot scheduling problem.

TABLE 6. Mean and standard deviation results of HV obtained by involved algorithms on multi-robot scheduling problem.

Number of robots	MO-MOEA	MD-MOEA	IBEA
10	0.9711 (0.0080)	0.9618 (0.0051)	0.9484 (0.0196)
15	0.9926 (0.0023)	0.9667 (0.0063)	0.8135 (0.0337)

V. CONCLUSION

This paper formulated a multi-robot scheduling model in the intelligence warehouse and designed a novel maximin based algorithm to solve the model. This algorithm uses one-by-one update scheme together with additional performance indicators and similarity verification method to solve the dilemma of equivalent selection within traditional maximin function. Experimental results on a set of many-objective benchmarks with up to 15 objectives show the effectiveness of our proposed algorithm in solving MaOPs. Then our algorithm was applied in the multi-robot scheduling problem, and received promising results.

In the future, we will make further research on the efficiency of the maximin fitness function in solving practical problems. This approach will be used to resolve more complex real-world multi-robot scheduling problems with more objectives.

REFERENCES

[1] M. Alavidoost, M. Tarimoradi, and M. F. Zarandi, "Bi-objective mixed-integer nonlinear programming for multi-commodity tri-echelon supply chain networks," *J. Intell. Manuf.*, vol. 29, pp. 809–826, Jul. 2015.

[2] R. V. Barenji, A. V. Barenji, and M. Hashemipour, "A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop," *Int. J. Adv. Manuf. Technol.*, vol. 71, nos. 9–12, pp. 1773–1791, 2014.

[3] A. Costa, F. A. Cappadonna, and S. Fichera, "A hybrid genetic algorithm for minimizing makespan in a flow-shop sequence-dependent group scheduling problem," *J. Intell. Manuf.*, vol. 28, no. 6, pp. 1269–1283, 2017.

[4] M. A. Hariga and P. L. Jackson, "The warehouse scheduling problem: Formulation and algorithms," *IIE Trans.*, vol. 28, no. 2, pp. 115–127, Feb. 1996.

[5] S. Minner and S. Transchel, "Periodic review inventory-control for perishable products under service-level constraints," *OR Spectr.*, vol. 32, no. 4, pp. 979–996, Sep. 2010.

[6] J. Gu, M. Goetschalckx, and L. F. McGinnis, "Research on warehouse design and performance evaluation: A comprehensive review," *Eur. J. Oper. Res.*, vol. 203, no. 3, pp. 539–549, Jun. 2010.

[7] M. Petrović, Z. Miljković, and A. Jokić, "A novel methodology for optimal single mobile robot scheduling using whale optimization algorithm," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105520.

[8] A. Elmi and S. Topaloglu, "A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots," *Comput. Oper. Res.*, vol. 40, no. 10, pp. 2543–2555, Oct. 2013.

[9] G. Nastasi, V. Colla, S. Cateni, and S. Campigli, "Implementation and comparison of algorithms for multi-objective optimization based on genetic algorithms applied to the management of an automated warehouse," *J. Intell. Manuf.*, vol. 27, pp. 1545–1557, Jan. 2016.

[10] B. Xing, W.-J. Gao, F. V. Nelwamondo, K. Battle, and T. Marwala, "Ant colony optimization for automated storage and retrieval system," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, Jul. 2010, pp. 1–7.

[11] X. Chen, Y. Li, and L. Liu, "A coordinated path planning algorithm for multi-robot in intelligent warehouse," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 2945–2950.

[12] Z. Li, A. V. Barenji, J. Jiang, R. Y. Zhong, and G. Xu, "A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand," *J. Intell. Manuf.*, vol. 31, no. 2, pp. 469–480, Feb. 2020.

[13] F. Chen, H. Wang, Y. Xie, and C. Qi, "An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse," *J. Intell. Manuf.*, vol. 27, no. 2, pp. 389–408, Apr. 2016.

[14] T. Sandholm, "Algorithm for optimal Winner determination in combinatorial auctions," *Artif. Intell.*, vol. 135, nos. 1–2, pp. 1–54, 2002.

[15] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2003, pp. 1957–1962.

[16] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sendai, Japan, Sep./Oct. 2004, pp. 698–705.

[17] L. Ma, R. Wang, S. Chen, X. Wang, S. Cheng, and Y. Shi, "A novel many-objective evolutionary algorithm based on transfer matrix with Kriging model," *Inf. Sci.*, vol. 509, pp. 437–456, Jan. 2020.

[18] F. Hao, G. Min, M. Lin, C. Luo, and L. T. Yang, "MobiFuzzyTrust: An efficient fuzzy trust inference mechanism in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2944–2955, Nov. 2014.

[19] S. Cheng, L. Ma, H. Lu, X. Lei, and Y. Shi, "Evolutionary computation for solving search-based data analytics problems," *Artif. Intell. Rev.*, vol. 54, no. 2, pp. 1321–1348, Feb. 2021, doi: 10.1007/s10462-020-09882-x.

[20] S.-Z. Zhou, Z.-H. Zhan, Z.-G. Chen, S. Kwong, and J. Zhang, "A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 8, 2020, doi: 10.1109/TITS.2020.2994779.

[21] W. Fang, Q. Zhang, J. Sun, and X.-J. Wu, "Mining high quality patterns using multi-objective evolutionary algorithm," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 23, 2020, doi: 10.1109/TKDE.2020.3033519.

[22] J. Sun, H. Li, Y. Zhang, Y. Xu, Y. Zhu, Q. Zang, Z. Wu, and Z. Wei, "Multiobjective task scheduling for energy-efficient cloud implementation of hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 587–600, 2021.

[23] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. 8th Int. Conf. Parallel Problem Solving Nature (PPSN)*, Birmingham, U.K., 2004, pp. 832–842.

[24] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, "TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things," *IEEE Trans. Mobile Comput.*, early access, Mar. 8, 2021, doi: 10.1109/TMC.2021.3064314.

[25] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[26] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," in *Proc. 7th Int. Conf. Learn. Intell. Optim.*, Catania, Italy, 2013, pp. 70–74.

[27] R. H. Gómez and C. A. Coello Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *Proc. Annu. Conf. Genetic Evol. Comput.*, Madrid, Spain, Jul. 2015, pp. 679–686.

[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[29] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. 5th Conf. Evol. Methods Design Optim. Control Appl. Ind. Problems*, Athens, Greece, 2001, pp. 95–100.

[30] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.

[31] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[32] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[33] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[34] R. Balling, "Pareto sets in decision-based design," *J. Eng. Valuation Cost Anal.*, vol. 3, pp. 189–198, Mar. 2000.

[35] L. Ma, K. Hu, Y. Zhu, and H. Chen, "Cooperative artificial bee colony algorithm for multi-objective RFID network planning," *J. Netw. Comput. Appl.*, vol. 42, pp. 143–162, Jun. 2014.

[36] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, "Hyperplane assisted evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3367–3380, Jul. 2020.

[37] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, Apr. 2018.

[38] Y. Tian, X. Zhang, R. Cheng, and Y. Jin, "A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 5222–5229.

[39] L. Ma, X. Wang, M. Huang, Z. Lin, L. Tian, and H. Chen, "Two-level master–slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 861–880, May 2019.

[40] A. Menchaca-Mendez and C. A. Coello Coello, "Selection operators based on maximin fitness function for multi-objective evolutionary algorithms," in *Proc. Evol. Multi–Criterion Optim., 7th Int. Conf., (EMO)*, Sheffield, U.K., Mar. 2013, pp. 215–229.

[41] R. Balling, "The maximin fitness function; multi-objective city and regional planning," in *Proc. 2nd Int. Conf. (EMO)*, Faro, Portugal, Apr. 2003, pp. 1–15.

[42] A. Menchaca-Mendez and C. A. Coello Coello, "Solving multi-objective optimization problems using differential evolution and a maximin selection criterion," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–8.

[43] A. Menchacamendez and C. A. C. Coello, "MD-MOEA: A new MOEA based on the maximin fitness function and Euclidean distances between solutions," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 2148–2155.

[44] R. Balling and S. Wilson, "The maximin fitness function for multi-objective evolutionary computation: Application to city planning," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, San Francisco, CA, USA, 2001, pp. 1–9.

[45] H. Chen, L. Ma, M. He, X. Wang, and M. Huang, "Artificial bee colony optimizer based on bee life-cycle for stationary and dynamic optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 327–346, Feb. 2017.

[46] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689–2702, Sep. 2017.

[47] K. Yu, J. Liang, B. Qu, Y. Luo, and C. Yue, "Dynamic selection preference-assisted constrained multiobjective differential evolution," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Mar. 9, 2021, doi: 10.1109/TSMC.2021.3061698.

[48] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 233–246, Jan. 2020.

[49] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[50] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[51] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, Mar. 2017.

[52] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jan. 22, 2020, doi: 10.1109/TSMC.2020.2963943.

[53] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[54] L. Ma, S. Cheng, X. Wang, M. Huang, H. Shen, X. He, and Y. Shi, "Cooperative two-engine multi-objective bee foraging algorithm with reinforcement learning," *Knowl.-Based Syst.*, vol. 133, pp. 278–293, Oct. 2017.

[55] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," *Evol. Multiobjective Optim.*, pp. 105–145, 2005.

[56] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[57] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA, USA: Springer, 2007, pp. I–XXI and 1–800.

[58] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, early access, Jan. 21, 2021, doi: 10.1109/TCYB.2020.3041212.

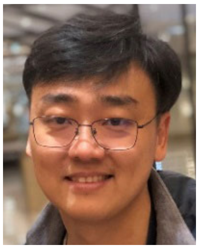
[59] K. Deb, "Multi-objective optimization using evolutionary algorithms," in *Proceedings of Wiley-Interscience Series in Systems and Optimization*. New York, NY, USA: Wiley, 2001, pp. 1–497.

[60] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inform.*, vol. 26, no. 4, 30-45, 1996.

[61] G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 542–555, Feb. 2019, doi: 10.1109/TCYB.2017.2780274.

[62] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

SHUJUN YANG received the B.S. degree in network engineering from Shenyang Normal University, China, in 2018. He is currently pursuing the M.S. degree with Northeastern University, China. His current research interest includes multi-objective evolutionary algorithms.



YICHUAN ZHANG received the B.Sc. degree in computer science and technology from Liaoning University, Shenyang, Liaoning, China, in 2003, and the combined master's and Ph.D. degree in computer application technology from Northeastern University, Shenyang, in 2011. He is currently a Lecturer with Northeastern University. His current research interests include cloud computing, big data, and machine learning.

LIANBO MA received the B.Sc. degree in communication engineering and the M.Sc. degree in communication and information system from Northeastern University, Shenyang, China, in 2004 and 2007, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, China, in 2015. He is currently a Professor with Northeastern University. His current research interests include computational intelligence and machine learning.

YAN SONG received the M.S. degree in computer science and the Ph.D. degree in mechatronic engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, China, in 2007 and 2013, respectively. He worked as an Expert of IEC/SC65C/MT9 and an Expert of TC124, China. He is currently an Associate Professor with Liaoning University, China. His research interests include industrial communication, industrial security, and functional safety.

PING ZHOU received the B.S. degree in software engineering from Hebei University of Engineering, China, in 2019. She is currently pursuing the M.S. degree with Northeastern University, China. Her current research interests include evolutionary algorithms and information centric networks.

GANG SHI has been working with Shenyang Institute of Automation, Chinese Academy of Sciences, since July 2001. He currently presides over a number of research projects at the national, academy, provincial, and municipal levels, and horizontal projects. His past main work and achievements presided over two national-level projects, one provincial-level project, two academic-level projects, and two horizontal projects. He published more than ten academic articles, of which ten were indexed by EI/SCI/ISTP/SSCI. He received the First Prize of Shenyang Science and Technology Progress Award and the Knowledge Innovation Project Crown two famous awards.

HANNING CHEN received the B.S. degree in mechanical design and automation and the M.S. degree in computer software and theory from Yanshan University, Qinhuangdao, China, in 2002 and 2005, respectively, and the Ph.D. degree from the Chinese Academy of Sciences, China, in 2009. He is currently a Professor with Tiangong University. His current research interest includes biologically inspired computation.