sustainability

MDPI

Article

Research on Multi Unmanned Aerial Vehicles Emergency Task Planning Method Based on Discrete Multi-Objective TLBO Algorithm

Miao Tang [1,*], Minghua Hu [1], Honghai Zhang [1] and Long Zhou [1,2]

[1] College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; minghuahu@nuaa.edu.cn (M.H.); honghaizhang@nuaa.edu.cn (H.Z.); zhoulongzwzl@163.com (L.Z.)
[2] Nanjing Intelligent Aviation Research Institute Co., Ltd., Nanjing 210007, China
* Correspondence: tangmiao@nuaa.edu.cn

Abstract: The outbreak of unexpected events such as floods and geological disasters often produces a large number of emergency material requirements, and when common logistics methods are often ineffective, emergency logistics unmanned aerial vehicles (UAVs) become an important means. How to rationally plan multiple UAVs to quickly complete the emergency logistics tasks in many disaster-stricken areas has become an urgent problem to be solved. In this paper, an optimization model is established with the goal of minimizing the task completion time and the penalty cost of advance/delay, and a discrete multi-objective teaching–learning-based optimization (DMOTLBO) algorithm is proposed. The Pareto frontier approximation problem is transformed into a set of single objective sub-problems by the decomposition mechanism of the algorithm, and each sub-problem is solved by the improved discrete TLBO algorithm. According to the characteristics of the problem, TLBO algorithm is improved by discretization, and an individual update method is constructed based on probability fusion of various mutation evolution operators. At the same time, variable neighborhood descent search is introduced to enhance the local search ability. Based on the multi-level comparative experiment, the improvement measures and effectiveness of DMOTLBO are verified. Finally, combined with specific case analysis, the practicability and efficiency of the DMOTLBO algorithm in solving the multi-objective emergency logistics task planning problem of multiple UAVs are further verified.

Keywords: task planning; emergency logistics UAV; discrete multi-objective; improved TLBO algorithm

Citation: Tang, M.; Hu, M.; Zhang, H.; Zhou, L. Research on Multi Unmanned Aerial Vehicles Emergency Task Planning Method Based on Discrete Multi-Objective TLBO Algorithm. *Sustainability* **2022**, *14*, 2555. https://doi.org/10.3390/su14052555

Academic Editors: Xiaobei Jiang, Haixiang Lin, Fei Yan and Qian Cheng

Received: 2 January 2022
Accepted: 21 February 2022
Published: 23 February 2022

1. Introduction

Emergencies such as floods and geological disasters often generate a large number of emergency needs for emergency supplies. It is necessary to deliver emergency supplies to the disaster areas quickly, efficiently and accurately in a reasonable and feasible way, so as to meet the needs of basic survival, treatment of the wounded, sanitation, epidemic prevention, etc. However, under the complicated terrain conditions and bad weather environment in disaster areas, it is difficult to realize timely and effective emergency logistics support, since the commonly used logistics methods are often ineffective. In recent years, unmanned aerial vehicles (UAVs) have been applied in various places to make up for the shortage of emergency logistics [1]. How to rationally plan multiple UAVs in the base to quickly complete the emergency logistics tasks in many disaster-stricken areas has become an urgent problem to be solved.

At present, the research on UAV task planning is mostly transformed into combinatorial optimization problems, and the traditional solutions are deterministic method based on mixed integer programming (MILP), task allocation method based on market mechanism and dynamic network flow optimization method. The first one is to model the UAV task assignment as an integer programming problem, which can be solved by branch definition method and tangent plane method, which are two common deterministic algorithms. However, the integer programming algorithm requires the number of UAVs exceeds that of targets; furthermore, the cost calculation is too simple, so it is not suitable for the target allocation problem in practical common situations. Chang [2], Kim [3] and others analyzed the limitations of the classic multi-UAV task allocation model and established a multi-UAV task allocation model based on agent and contract network negotiation. The method is simple in principle, easy to implement and high in efficiency, but it has poor ability to deal with coordination and constraints, and probably conflicts with individual interests in pursuit of overall optimization. Sieatkowska [4] and Fu.Z [5] have constructed a dynamic network flow model with limited capacity, which can solve the optimal resource allocation problem of UAV. However, in order to construct a group of one-stage problem models, the methods oversimplify the cooperative relationship between UAVs and reduces the credibility. Moreover, the rapid development of intelligent optimization algorithms in recent years provides a new way to solve UAVs task planning problems, among which population-based algorithms are common. Population-based intelligent algorithms generally use the population composed of multiple solutions as the planning object, and through repeated iteration, find the optimal solution in the search space [6]. Commonly used swarm intelligence algorithms include genetic algorithm, particle swarm optimization algorithm, ant colony algorithm [7–12]. They are flexible, adaptive and inspiring; however, they only focus on the task planning and optimization of multiple UAVs under a single target, without considering the complexity of mission planning objectives in the actual environment. Additionally, the solution of multi-objective optimization problem is to seek the trade-off between multiple objectives of different dimensions, in which Pareto dominance-based method [13] has been widely used. Zhang Qingfu and others put forward the multi-objective evolutionary algorithm MOEA/D [14] based on decomposition in 2007, and introduced the decomposition method in mathematical programming into the field of multi-objective evolution. However, the multi-objective task assignment problem of multi-UAVs belongs to a complete NP-hard problem. The multi-objective nature of the solution process and the large number of UAVs involved will lead to the phenomenon of combined explosion, which further aggravates the difficulty of solution and needs the support of effective task assignment algorithm. In addition, most studies consider the task allocation of military UAV, but there are few studies on the task planning in the field of civilian and emergency rescue UAV logistics. Actually, combining mathematical programming methods with intelligent algorithms is a new idea to solve emergency UAVs multi-objective task allocation problems.

In this paper, the DMOTLBO algorithm, combining MOEA/D and improved discrete TLBO algorithm, is designed to solve the multi-objective task planning and scheduling problem of emergency UAVs. The TLBO algorithm is an efficient and intelligent optimization algorithm proposed by Rao and other scholars. Inspired by teaching behavior, this method realizes iterative evolution by simulating the phenomenon of teachers' classroom teaching and students' mutual learning, and has the advantages of less control parameters and fast convergence speed [15]. Literature research shows that the research of DMOTLBO algorithm for UAV task planning is non-existent. Firstly, the multi-objective Pareto frontier approximation problem is transformed into a set of single-objective sub-problems by the decomposition mechanism, and then the sub-problems are solved based on the improved discrete TLBO algorithm. On the premise of maintaining the updating mechanism of the standard TLBO algorithm, the algorithm is discretized, and the teaching and learning stages are improved, respectively, so that the algorithm can directly conduct global search based on TLBO idea in discrete solution space. In addition, a variable neighborhood descent search is constructed to greatly enhance the local search ability of the algorithm. Finally, through a series of simulation experiments, the feasibility and efficiency of DMOTLBO algorithm are verified.

2. Mathematical Modeling

2.1. Assumption

The cooperative task allocation of emergency rescue for multi-UAVs refers to assigning tasks to each UAV, determining the set of target locations of each UAV, the amount of emergency materials corresponding to each target point and the execution order of transporting materials, so as to achieve the highest overall efficiency of multi-UAVs in performing tasks [16].

In the scene of emergency logistics replenishment by using UAV after natural disasters, there are some disaster sites that are not far from each other but difficult to reach due to geographical or meteorological factors, and it is necessary to use UAVs of different types in the emergency rescue flight base to carry out transportation tasks with corresponding emergency materials such as medicines according to the degree of urgency. Suppose an emergency command center O undertakes the mission to deliver emergency rescue materials, mainly medicines and lightweight tools, to n disaster sites in a certain area which are geographically close but scattered and difficult to reach quickly by conventional vehicles in a short time. The center is now equipped with m various UAVs for emergency rescue, each of which is represented as $U_k(k = 1, 2, \ldots, m)$ and the maximum load and endurance time of different types of UAVs are different. Each UAV is assumed to be a particle with a constant velocity, that is, the dynamic characteristics of the UAV are not considered, and only the kinematic characteristics are taken into account. Furthermore, because the transportation distance considered in this paper is relatively short, the complicated environmental factors are not considered for the time being. Then, it is assumed that the flight distance between two locations of the UAV is close to the straight line distance between them, and the round-trip time of the UAV between the two locations is the same. Additionally, due to the limitation of energy, the UAV can only fly continuously for a limited distance.

First, the UAV completes the material transportation task of disaster site i, that is, completes the task m_{ki}. Assuming that the task execution set of UAV U_k is M_k, the corresponding total flight distance of U_k is L_k, and the UAVs are required to return to the base after completing the task set, then L_k should be the total flight length of U_k returning to the base from the last target point after carrying out emergency material delivery tasks of each target point in a given order. $L_k^{\max}$ denotes the maximum flight distance of U_k in a single flight. $Q_k^{\max}$ denotes the maximum load of U_k. Additionally, there are several constraints.

2.2. Constraints

Constraint (1): A single UAV can fly to single or multiple disaster sites. However, the round-trip distance of a single UAV flying mission is less than the maximum cruising distance of UAV.

$$L_k \leq L_k^{\max} \tag{1}$$

Constraint (2): It is assumed that the types and corresponding quantities of materials needed at each disaster site are known. q_i denotes the material demand corresponding to the task target point i, $Q(M_k)$ is the total material demand of all target points in U_k's task set, that is, the actual load of U_k when it leaves, so $Q(M_k)$ should not exceed $Q_k^{\max}$. During the transportation, the load decreases with the increase in pick-up times, and it is ignored for the little influence on flight performance in this paper.

$$Q(M_k) \leq Q_k^{\max} \tag{2}$$

Constraint (3) is used to ensure that each task is only executed and completed once.

$$\sum_{a=0, a\neq b}^{n} \sum_{k=1}^{m} x_{abk} = 1, b = 1, 2, \ldots, n \tag{3}$$

Constraint (4) shows each task is executed first or immediately after a certain task.

$$\sum_{a=0,a\neq s}^{n} x_{ask} - \sum_{b=1,b\neq s}^{n} x_{sbk} = 0,\ s = 1,2,\ldots,n,\ k = 1,2,\ldots,m \tag{4}$$

Constraint (5) indicates the completion time of each task. UAVs required a different amount of time to reach each target point in the disaster area, and it costs one UAV a different amount of time to complete transportation tasks of various sites.

$$\begin{gathered} T_b = \max\{T_a + \sum_{k=1}^{m} x_{abk} \cdot t_{abk} + s_{bk} + R \cdot (\sum_{k=1}^{m} x_{abk} - 1)\}, \\ a = 0,1,\ldots,n,\ b = 1,2,\ldots,n \end{gathered} \tag{5}$$

Since R represents a positive number large enough, the above formula can guarantee that b is the next target task point of a.

Constraint (6) indicates that the first mission of each UAV is no more than one.

$$\sum_{b=1}^{n} x_{0bk} \leq 1, k = 1,2,\ldots,m \tag{6}$$

Formula (7) represents the initialization of the task, and the completion time of virtual task 0 is 0, that is, the initial time from the base is 0.

$$T_0 = 0 \tag{7}$$

Formula (8) calculates the advance and delay time of each task.

$$\begin{gathered} TD_b \geq T_b - E_b, b = 1,2,\ldots,n \\ TD_b \geq T_b - E_b, b = 1,2,\ldots,n \end{gathered} \tag{8}$$

Formula (9) calculates the value of UAV mission planning and scheduling scheme.

$$T_{\max} = \max(T_b), b = 1,2,\ldots,n \tag{9}$$

Constraint (10) defines the value range of all variables.

$$\begin{gathered} x_{abk} \in \{0,1\},\quad a = 0,1,\ldots,n,\ \ b = 1,2,\ldots,n,\ \ k = 1,2,\ldots,m, \\ TF_b \geq 0, TD_b \geq 0, T_b \geq 0, b = 1,2,\ldots,n \end{gathered} \tag{10}$$

2.3. Objective Function

Additionally, before an UAV carries out the task, the base has already determined the expected pick-up time with each disaster-stricken point. When the UAV arrives before the available time, it needs personnel to wait for picking-up in advance, and when arrives late, it may cause problems such as delaying the timing of drug treatment and so on. Therefore, it is required that all UAVs should have the shortest total flight time and arrive on time as much as possible, that is, the least advance and delay.

The optimization objectives are as follows:

(1) Find the minimum value of the completion time of all tasks, that is, calculate with the UAV in the base with the longest completion time and find the minimum value:

$$\min f_1 = T_{\max} \tag{11}$$

(2) Minimizing the sum of advance/delay penalty costs:

$$\min f_2 = \sum_{b=1}^{n} (F_b \cdot TF_b + D_b \cdot TD_b) \tag{12}$$

The decision variables are:

$$x_{abk} = \begin{cases} 1, if\ b\ is\ the\ next\ task\ poin\ t\ after\ a\ of\ UAV\ U_k \\ 0, else \end{cases}$$
$$x_{0bk} = \begin{cases} 1, if\ b\ is\ the\ first\ task\ poin\ t\ for\ U_k \\ 0, else \end{cases} \quad (13)$$

The parameters' meanings are as follows:

a and b represent different disaster sites. m represents the total number of UAVs. n indicates the total number of UAV task target points that need material support. t_{abk} indicates the length of time for the UAV U_k to arrive at the disaster-stricken task point a from the task point b. l_{abk} denotes the flight distance of U_k from point a to point b. Assuming that U_k flies at a constant speed v_k, then $l_{abk} = t_{abk} \cdot v_k$. s_{bk} indicates the stay time of U_k in a certain task point b. As in this paper the stay time of picking up supplies is relatively short to the flight time, so s_{bk} is approximately zero. T_b indicates the delivery arrival time of task point. E_b denotes the expected delivery time of task point b. TF_b indicates the advance arrival time of task point b. TD_b indicates the delay arrival time of point b. F_b indicates the cost coefficient of advance punishment for task point b. D_b indicates the delay penalty cost coefficient of task point b. $T_{\max}$ indicates the maximum completion time of all tasks.

3. UAV Emergency Task Planning Based on the DMOTLBO Algorithm

3.1. TLBO Algorithm

Teaching–learning-based optimization algorithm TLBO is a new optimization method based on the classroom teaching effect proposed by Rao et al. It is a swarm intelligence evolutionary algorithm that simulates the teaching and influence of teachers on students in the classroom and the mutual learning process among students, and makes the whole group continuously evolve forward. Additionally, the group of teachers and students are the population in TLBO algorithm. The best individual in each generation becomes the teacher, and the rest are students. TLBO algorithm consists of two stages, namely, the teaching stage and the learning stage. The former is the stage when students learn from teachers, and the latter is the stage when students learn from each other to improve their grades. Given the population size N and the coding length D of the problem, and $X_i = [x_{i1}, x_{i2}, x_{i3} \ldots, x_{id}](i = 1, 2, \ldots, N; d = 1, 2, \ldots, D)$ denotes the i-th student in the class, x_{id} denotes the numerical value of student X_i in dimension d, indicating the achievement of a certain course. The upper and lower limit of achievement, that is, the range of independent variables of each dimension is $x_{id} = \left[x_{id}^l, x_{id}^u\right]$. The initial population formula is as follows and $r = rand(0,1)$ denotes the learning step, which is a random number on [0, 1].

$$X_i = L + r \cdot (U - L), L = \left(x_{i1}^l, x_{i2}^l, x_{i3}^l, \ldots, x_{id}^l\right), U = (x_{i1}^u, x_{i2}^u, x_{i3}^u, \ldots, x_{id}^u) \quad (14)$$

3.1.1. Teaching Stage

As a teacher, the optimal individual in the class population X_t updates the population through the "teaching" operator. Given the parent individual, the formula for generating new individuals is shown as below:

$$X_{inew}(t) = X_{iold}(t) + r \cdot (X_t(t) - T_F \cdot X_m(t)); X_m(t) = \{m_1(t), m_2(t) \ldots, m_d(t)\} \quad (15)$$

in which t is the current iteration number, and $m_d(t)$ is the average score in each course of all t-generation students, that is, the average value of independent variable in each dimension currently.$X_t(t)$ means the optimal individual found in the t-generation, which is also the expected average level of the next generation. Additionally, as the teaching factor, $T_F = round(1 + rand(0,1)), T_F \in \{1, 2\}$. $X_{inew}(t)$ and $X_{iold}(t)$ denote the i-th individual before and after the update in the t-generation. At last, comparing the objective or fitness function values of the two, the current learning result will be accepted only if it is better after updating.

3.1.2. Learning Stage

In this stage, the "learning" operator is used to realize mutual learning among individuals, that is, randomly select individuals in the population, and update the current population with the difference component between the individual and other individual vectors for the second round. Taking the minimization problem as an example, using $f(X_i)$ to represent the current optimization problem the objective function, the formula for generating new individuals in the learning stage is as follows:

$$X_{inew}(t) = \left\{ \begin{array}{l} X_{iold}(t) + r\cdot\left(X_j(t) - X_{iold}(t)\right), f\left(X_j(t)\right) < f\left(X_{iold}(t)\right) \\ X_{iold}(t) + r\cdot\left(X_{iold}(t) - X_j(t)\right), f\left(X_j(t)\right) \geq f\left(X_{iold}(t)\right. \end{array} \right\} \tag{16}$$

Compare the corresponding objective function or fitness values of $X_{inew}(t)$ and $X_{iold}(t)$, then take the better solution as the offspring individual.

The standard TLBO algorithm has a simple parameter model, fast convergence speed and strong search ability, but it is easy to fall into local optimum because of less population diversity. Therefore, this paper improves the standard TLBO algorithm by the DMOTLBO algorithm, introducing the idea of discretization and a mutation operator based on probability.

3.2. UAV Emergency Task Planning Based on DMOTLBO Algorithm

The DMOTLBO algorithm adopts a decomposition mechanism and a set of different weight vectors to decompose the multi-objective optimization problem (MOP) into a set of single-objective optimization problems for simultaneous solution, optimizing each sub-problem with TLBO algorithm. Each sub-problem divides neighbors according to its own weight vectors, and employs the co-evolution mechanism between sub-problems to improve the information sharing of neighbor solutions and reduce the computational complexity. In order to ensure the efficient operation of the algorithm, combining with the characteristics of UAV task assignment problem, a sequence coding method is designed; based on this, improved discrete teaching and learning stages are applied to the individual evolution process, and a variable neighborhood descent search stage is added to strengthen local search.

3.2.1. Decomposition Mechanism

For small-scale examples, the ε constraint method is used to transform a certain objective in the bi-objective optimization model into a constraint. By constructing a set of single-objective ε constraint problems and accurately solving them with CPLEX, the Pareto optimal frontier of the current bi-objective task allocation problem is obtained. The specific process is shown in the figure below, Ω and ε represent the search space of the problem and a small positive number, respectively.

For medium- and large-scale examples, all optimized non-inferior solutions are regarded as approximate Pareto optimal frontier.

At the same time, the multi-objective UAV task planning problem is divided into N subproblems, where N is equal to the population size, and the weight vector is designed with uniform mixture. The weight vectors corresponding to subproblems are set as follows: $w_i = (\lambda_i^1, \lambda_i^2)$ in which λ_i is indicated as: $\lambda_i = (\frac{i-1}{N-1}, \frac{N-i}{N-1}), i = 1, 2, \ldots, N$. The objective function of subproblem is set by normalized Chebyshev aggregation method, and the operation mechanism is shown in Figure 1.

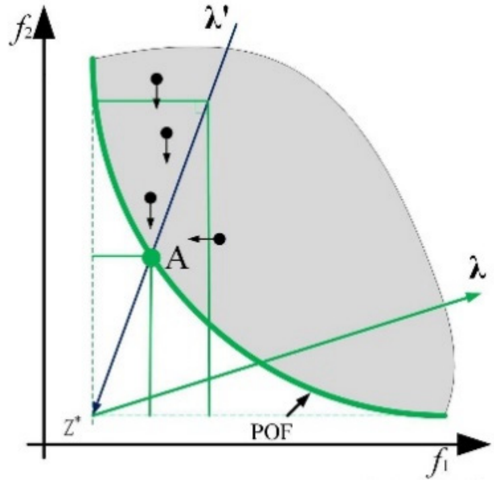

Figure 1. Schematic diagram of Chebyshev aggregation method.

For sub-problem i, $F_i(x)$ is used to express the fitness of solution x. f_1 and f_2 represent the values of Objective 1 and 2 of solution x. $f_1^{\max}, f_1^{\min}, f_2^{\max}, f_2^{\min}$ represent the maximum and minimum values of Objective 1 and 2 under the current iteration times, respectively. Additionally, if $Z^* = (Z_1^*, Z_2^*) = (0,0)$ is the reference point, then the aggregation function can be expressed by the following Formula.

$$F_i(x) = \max[\lambda_i^1 \cdot (\frac{f_1 - f_1^{\min}}{f_1^{\max} - f_1^{\min}} - O_1), \lambda_i^2 \cdot (\frac{f_2 - f_2^{\min}}{f_2^{\max} - f_2^{\min}} - O_2)] \quad (17)$$

According to the above formula, combined with the Tchebycheff aggregation mechanism, the algorithm will search the intersection point of each weight vector and Pareto frontier in the feasible solution space. Because of the uniform distribution of λ_i, the algorithm will obtain a group of uniform solutions on Pareto frontier. In addition, the Euclidean distance between weight vectors is calculated, and the nearest T weight vectors are taken as neighbors of each sub-problem.

3.2.2. Sequence Coding Mode

TLBO algorithm was originally used to solve the continuous variable optimization problem, and all individuals in the population were coded by real numbers. At present, for the research of UAV task planning, although the solution of the problem can be obtained through reasonable decoding rules by using real coding, the search efficiency of the algorithm is low because of the redundant information. Therefore, sequence coding is used to represent the solution of UAV task planning problem. Each individual is a solution to the problem. Given n denotes total number of tasks when m indicates the total number of UAVs, the code length is ($n + m - 1$), where code 1~n represent the numbers of task target points, and code ($n + 1$)~($n + m - 1$) are the division symbols. From this, it can be seen that ($m - 1$) separators can divide the arrangement of Task 1~n into m subsequences (including empty sequences), which constitute the task sequences of the corresponding UAV. Assuming that the total number of tasks and UAVs are 12 and 5, respectively, Figure 2 shows the encoding and decoding method of the example, where the coded sequence (12, 10, 8, 11, 14, 6, 9, 16, 1, 5, 13, 3, 7, 15, 2, 4), code 1–12 are the task numbers, and code 13–16 represent the division symbol, which can be used to obtain the disaster-stricken points that each UAV needs to perform the material delivery task.

Encoding:

12	10	8	11	**14**	6	9	**16**	1	5	**13**	3	7	**15**	2	4

Figure 2. An example of sequence coding mode.

After decoding, it means that: UAV 1 carries out the flight task M_1, including disaster sites 12, 10, 8 and 11, when UAV 2's mission is M_2, including task target points 6 and 9, and UAV 3's flying mission M_3 includes target points 1 and 5, meanwhile UAV 4's task set is M_4, including points 3 and 7; UAV 5's task set is M_5 which includes task target points 2 and 4. Additionally, UAVs need to execute subtasks in order.

3.2.3. Adaptive Discrete Teaching Stage

The main teaching stages are divided into teaching preparation, teacher training and teaching stage. In the early stage of the algorithm, the population mainly move closer to the optimal individual quickly to learn from the teacher. However, as the iteration goes on, the ability of individuals to maintain their own state is enhanced, which slows down the speed of approaching the optimal individual and avoids gathering around the teacher prematurely. For each individual in the population in the current iterative state $X_i(t)$, the update is realized by discretization on the basis of the teaching stage of the standard TLBO algorithm, and the specific operation is carried out according to the following formula.

$$X_i(t+1) = OBX\{\delta \otimes X_{iold}(t), T_F \otimes PMX[X_t(t), X_m(t), m, n], m', n'\} \tag{18}$$

In teaching preparation stage, the preview process of students before class in the current iterative state is represented by $\delta \otimes X_i(t)$, that is, the dynamic adaptive learning of student individuals in the teaching stage. Additionally, nonlinear adaptive mutation factor δ and random number r are introduced to perform mutation operation, in which $\delta = \gamma[\cos(\pi \cdot \frac{t}{T}) + \lambda], \quad \lambda = 1, \quad \gamma = 0.5, \quad \delta \in (0,1)$, λ is the value step of δ when γ is δ's change rate and $r = rand(0,1)$. Only when $r \leq \delta$ the mutation operation is conducted. It can be seen that with the increase in iteration times, students' ability to maintain their own state is enhanced, which slows down the speed of approaching the optimal individual so avoids gathering around the teacher prematurely.

Additionally, three neighborhood operations, namely, exchange, insert and changeover, are designed to achieve the mutation effect after preview as shown in Figure 3. Two different integers, i and j, are randomly generated, which are not greater than the encoding length, given that $Exchange(X_i, i, j)$ is exchanging the code at the i-th position in Solution X_i with that at the j-th position to generate a new solution, and the update of solution with $Insert(X_i, i, j)$ will insert the code at the i-th position in X_i into the j-th position, when during the process of $Changeover(X_i, i, j)$ the code between the i-th and j-th positions can be reversed.

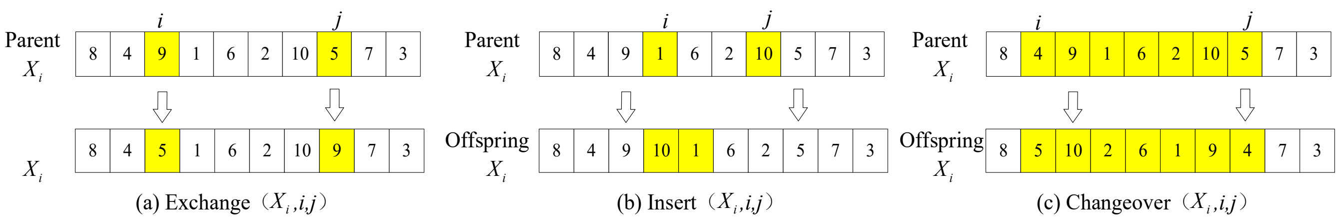

Figure 3. Schematic diagram of three mutation operators.

In order to better search different areas of the problem solution space, the whole iteration process is divided into three stages, and the above three neighborhood operations are performed, respectively. The self-learning process is shown in Formula (18):

$$X_{inew}(t) = \delta \otimes X_{iold}(t) = \begin{cases} Exchange[X_{iold}(t), i, j], if\ 0 < t \le T/3 \& r \le \delta; \\ Insert[X_{iold}(t), i, j], if\ T/3 < t \le 2T/3 \& r \le \delta; \\ Changeover[X_{iold}(t), i, j], if\ \ 2T/3 < t \le T \& r \le \delta; \\ X_{iold}(t), else. \end{cases} \tag{19}$$

In the teacher training stage, the population mean is updated discretely, which corresponds to the operation of finding the gap between the current optimal individual and the average level in the teaching stage of standard TLBO algorithm, and can also be understood as training and optimizing the teacher by the following formula.

$$X_{tnew}(t) = \begin{cases} T_F \otimes PMX[X_t(t), X_m(t), m, n], if\ \ T_F = 2 \\ X_m(t),\ else \end{cases} \tag{20}$$

Select an individual $X_t(t)$ randomly from the external archive EP to represent the teacher, when $X_m(t)$ represents the average score of the current population through iteration. Additionally, according to many tests, when T_F is 2, it is more effective to execute Partially Matching Crossover $PMX(\cdot)$. The workflow is shown in Figure 4a. Firstly, select consecutive coding positions between m and n ($m \le n$) in $X_t(t)$ and $X_m(t)$; secondly, the code selected in $X_m(t)$ is placed in the same position of $X_t(t)$ to generate temporary offspring individuals; finally, conflict detection is carried out, mapping relationship is established according to the code values at the selected positions, and the repeated codes in temporary offspring individuals are mapped to other codes, so as to generate a new expected average level as $X_{tnew}(t)$.

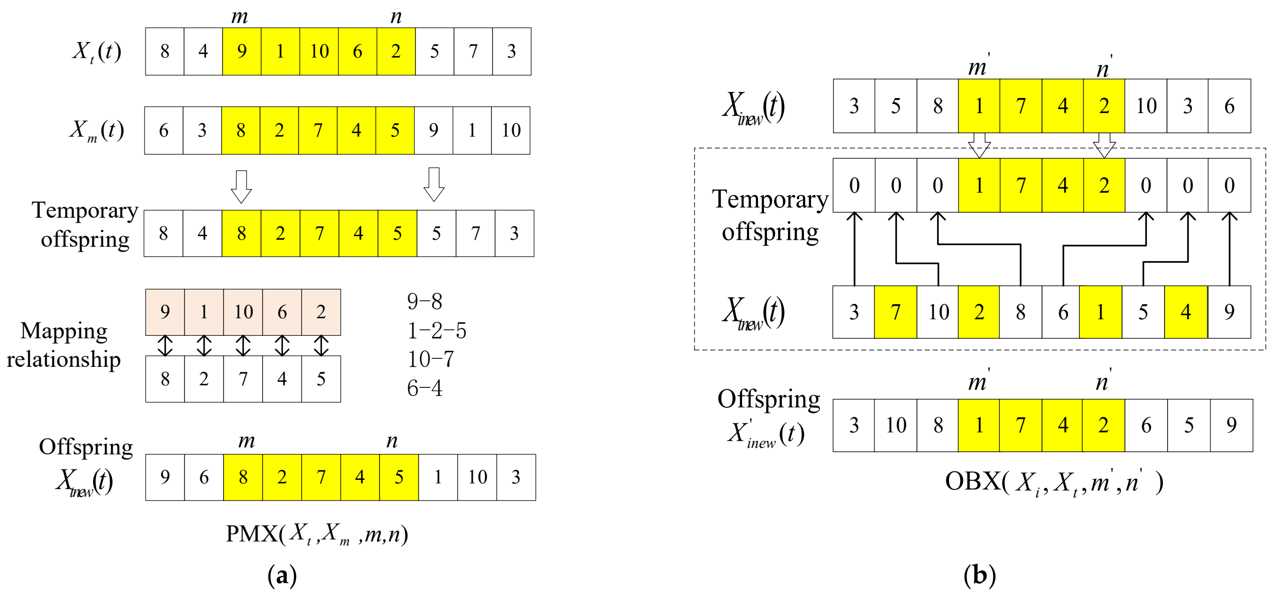

Figure 4. Schematic diagram of crossover operators, where (**a**,**b**) represent the schematic diagram of $PMX(\cdot)$ and $OBX(\cdot)$, respectively.

In the teaching stage, each individual continuously learns and then improves the average fitness value of the whole population through the order-based crossover operator as the following formula

$$X'_{inew}(t) = OBX\{X_{inew}(t), X_{tnew}(t), m', n'\}. \tag{21}$$

shown in Figure 4b. The steps are as follows: firstly, randomly select the sequential coding positions between m' and n' ($m' \leq n'$) in $X_{inew}(t)$ and $X_{tnew}(t)$, secondly, keep the selected codes in $X_{inew}(t)$ and set the rest to 0 to generate temporary offspring individuals, then determine the position in $X_{tnew}(t)$ of the non-zero codes in the temporary progeny and put the rest into the zero position of the temporary progeny in order to generate a new individual. Finally, compare the aggregate function values of $X_i(t+1)$ and $X_{iold}(t)$, and keep the better one as $X''_{inew}(t)$.

3.2.4. Discrete Learning Stage

First, use $X''_{inew}(t)$ from teaching stage to update $X_i(t)$. Additionally, the discretization update is realized according to the following formulas:

$$
\begin{aligned}
&X''_{inew}(t) = r \otimes PBX[X_i(t), X_i^{new}(t)] = \begin{cases} PBX[X_i(t), X_i^{new}(t)],\ if\ r > 0.5; \\ X_i^{new}(t), else \end{cases} \\
&X_i^{new}(t) = \begin{cases} OBX[X_i(t), X_j(t), p, q],\ if\ F_i(X_i(t)) < F_i(X_j(t)) \\ OBX[X_j(t), X_i(t), p, q],\ if\ F_i(X_i(t)) \geq F_i(X_j(t)) \end{cases}, r = rand(0,1)
\end{aligned} \tag{22}
$$

In the learning stage of the standard TLBO algorithm, student individuals learn from other students with a certain learning probability r. However, in DMOTLBO, students learn from each other through crossover operation. First of all, for $X_i(t)$, another individual $X_j(t)$ is randomly selected from its neighbors. $X_i^{new}(t)$ is generated by $OBX(\cdot)$ operation between $X_i(t)$ and $X_j(t)$ in the interval $[p, q]$. Owing to the update carried out among students in a small range, it can avoid premature gathering in the direction of global optimum and effectively ensure the diversity of population.

However, just like learning in real life, you also need to have a certain ability to identify what you have learned. If student individual s absolutely trusts and receives the acquired knowledge in the mutual learning stage, the algorithm may easily fall into local optimum. Therefore, the position-based crossover operator $PBX(\cdot)$ shown in Figure 5 is further introduced. When learning probability $r > 0.5$, randomly select multiple coding positions (which can be discontinuous) in $X_i(t)$, find the positions of the selected codes in $X_i^{new}(t)$, and set the rest to 0 to generate temporary children, then find out the positions of non-zero codes in temporary children, and put the rest of the codes in order into temporary children to replace 0 to generate new individuals $X'_{inew}(t)$. Finally, compare the aggregate function values of $X''_{inew}(t)$ and $X_i(t)$, and keep the better to update $X_i(t)$. Additionally, the execution of the whole DMOTLBO discrete search has ended till now.

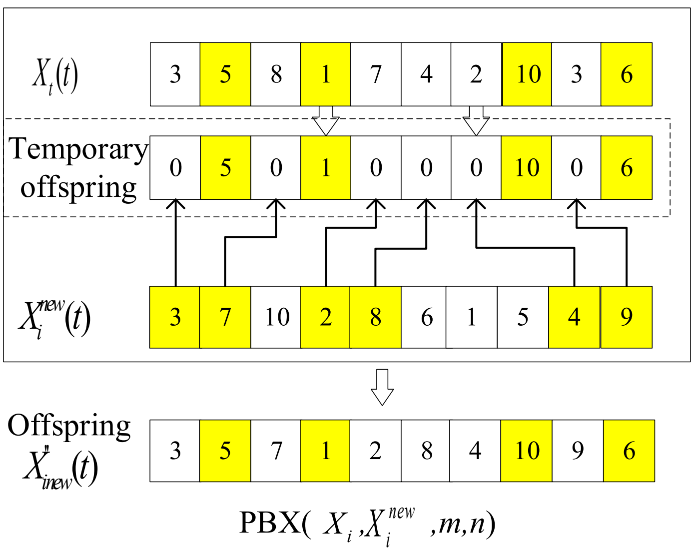

Figure 5. Schematic diagram of $PBX(\cdot)$.

3.2.5. Variable Neighborhood Search

Considering that the variable neighborhood descent search algorithm has strong local development ability, a corresponding stage is added to DMOTLBO. The main idea is to use multiple different neighborhoods for system search. First, the minimum neighborhood is used, and when the solution cannot be improved, it is switched to a slightly larger neighborhood. If it can continue to improve the solution, the algorithm workflow will return to the smallest neighborhood, otherwise will continue to switch to a larger neighborhood.

Specifically, the disturbance operation is performed once for each high-quality solution $X_i(t)$ in local search, then the insertion operator $Insert(\cdot)$ used to fulfill variable neighborhood descent search, the search depth is controlled by parameters, and X^* is used to record the optimal solution in the optimization process. The process is as follows.

Step1: do $X^* \leftarrow X_i(t)$, $l \leftarrow 1$, respectively, and turn to step 2;

Step2: perturb the current solution $X_i(t)$ by using the neighborhood structure exchange operator $Exchange(\cdot)$, and then generate the variation solution $tem1$, that is, $tem1 = Exchange(X_i,u,v)$. If $F_i\ (tem1) < Fi\ (X^*)$, do $X^* \leftarrow tem1$;

Step3: mutate the current solution $tem1$ by using the neighborhood structure insertion operator $Insert(\cdot)$, and then generate the mutated solution $tem2$, that is, $tem2 = Insert(tem1,u,v)$.

Step4: if $F_i\ (tem2) < F_i\ (X^*)$, make $tem1 \leftarrow tem2$, $X^* \leftarrow tem2$, $l \leftarrow 1$, respectively, then turn to step 3, otherwise, let $l \leftarrow l + 1$, and go to step 5.

Step5: if $l < L$, turn to step 3, otherwise, go to step 6.

Step6: terminate the iterative search.

Step7: update X^*, and do $X_i(t + 1)= X^*$.

Besides, u, v ($u \neq v$) are random numbers, and u, v are regenerated every time $Exchange(\cdot)$ and $Insert(\cdot)$ operations are performed.

3.3. DMOTLBO Algorithm Workflow

The workflow of the DMOTLBO algorithm is shown in Figures 6 and 7. At the initial stage of DMOTLBO, firstly generate N initial solutions x_i ($i = 1, 2, \ldots, N$) randomly, and then generate N weight vectors λ_i ($i = 1, 2, \ldots, N$) by uniform mixture method, in which the set consisting of T weight vectors closest to the vector λ_i is denoted as $V_i = \{\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_T}\}$, whose corresponding lower is denoted as $P_i = \{i_1, i_2, \ldots, i_T\}$. Then, assign a weight vector to each solution. Calculate the aggregate function values, namely the fitness values, select the non-inferior solutions, and establish the external archive EP as the non-inferior solution set.

Then, the discrete teaching, learning stage and variable neighborhood descent search in the DMOTLBO algorithm are employed to evolve and update the population. The algorithm framework is shown in Figure 7. Therefore, in fact, the non-inferior solution in the initial population is used to initialize EP. Then, in the iterative process, EP is updated according to Pareto dominance relation.

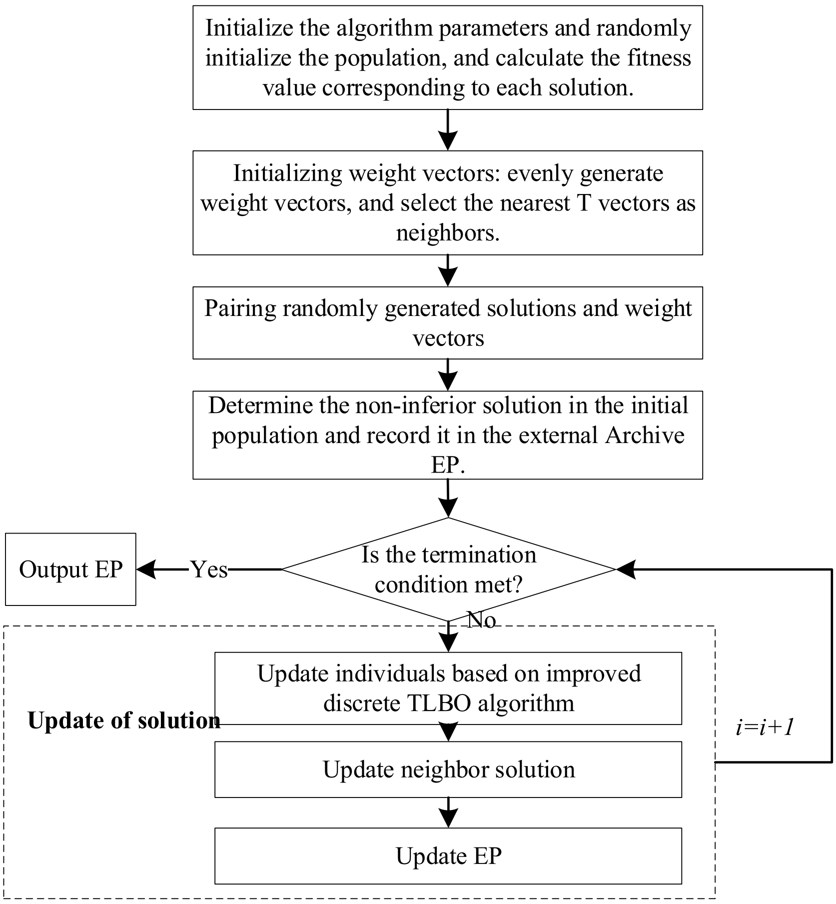

Figure 6. Workflow of the DMOTLBO algorithm.

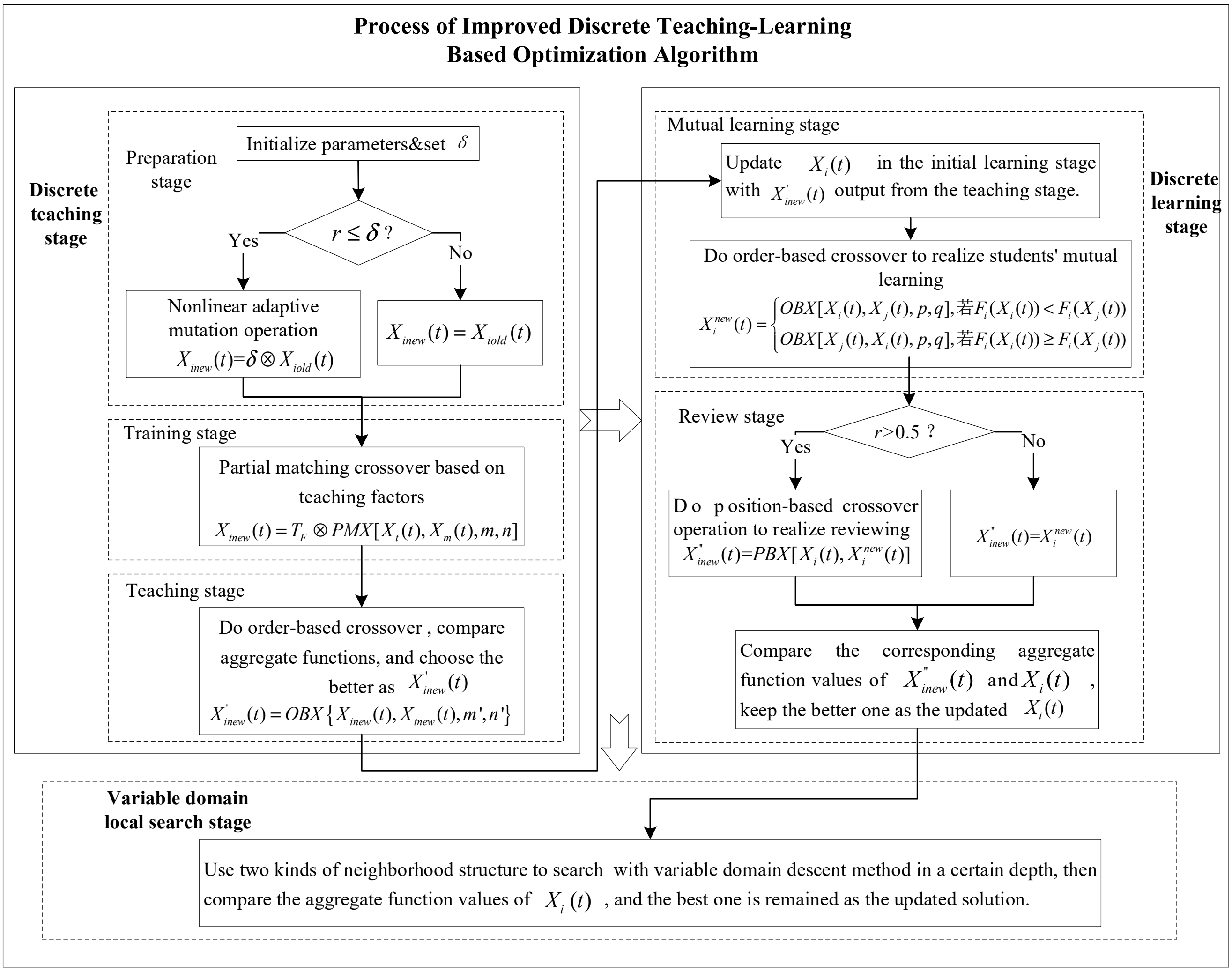

Figure 7. Workflow of the improved discrete TLBO algorithm.

4. Simulation and Analysis

4.1. Simulation Environment

The simulation was carried out on a computer with 16 GB memory and 11th Gen Intel (R) Core (TM) i7-1165G7 @ 2.80 GHz CPU, and MATLAB R2016a was used to program each test. Consult the Reference [17] and set small-scale and medium–large-scale examples to generate the test data set of this paper. Given the number of UAVs $m = 2$, set the number of task target points $n \in \{10, 12, 14, 16, 18, 20\}$; and if $m \in \{5, 8, 10\}$, then set $n \in \{30, 50, 80, 100, 150, 200\}$. The symbol $m \times n$ is used to represent cases of different scales, and 24 groups of cases are generated. The related parameters are set as follows: the coordinates of the task target point x, y are randomly generated in 60×60, the cost coefficient of advance penalty is $F_b \in \{0.1, 0.2, \ldots, 0.5\}$, when the cost coefficient of delay penalty is $D_b \in \{0.6, 0.7, \ldots, 1.0\}$, and the expected delivery time of materials obeys discrete uniform distribution $E_b \in DU(0, \frac{2}{5} \cdot \sum_{k=1}^{m} \sum_{b=1}^{n} \frac{l_{abk}}{m \cdot v_k})$. Besides, based on experiments, the algorithm can show good performance when the size of population and external archive are both set to 30. The maximum running time of different algorithms for solving small-scale examples is set to 20 s, and that of medium and large-scale examples is set to 60 s. The neighbor size t of the DMOTLBO algorithm and the iteration number LS of variable neighborhood search are set to 15 and 8, respectively.

The synthetic evaluation indicator Inverse Generation Distance (IGD), convergence indicator Generation Distance (GD) and uniformity indicator Spread (SP) are chosen to evaluate the algorithm performance.

(3) IGD is used to reflect the convergence and distribution of the algorithm. The smaller the IGD, the better the overall performance of the algorithm including convergence and distribution. IGD can be calculated by Formula (22).

$$IGD(P,P^*) = \frac{\sum\limits_{v \in P^*} d(v,P)}{|P^*|} \tag{23}$$

(4) GD is used to measure the average distance between each point in the non-inferior solution set and the real frontier. The smaller the value of GD, the better the convergence of the algorithm. Additionally, its calculation formula is:

$$GD(P,P^*) = \frac{\sum\limits_{x \in P} d(x,P^*)}{|P|} \tag{24}$$

(5) SP, which can be calculated by Formula (24), is used to measure the distribution uniformity of the non-inferior solution set. The smaller the SP, the better the performance of the algorithm.

$$SP = \frac{d_f + d_l + \sum\limits_{i=1}^{|P|-1} \left| d_i - \overline{d} \right|}{d_f + d_l + (|P|-1) \cdot \overline{d}} \tag{25}$$

In the formulas above, P^* is a set of uniformly distributed reference points sampled from the true Pareto frontier PF of the test problem, P is the Pareto solution set obtained, and $|P^*|$ is the number of individuals in the point set distributed on the real frontier. $d(v,P)$ represents the minimum Euclidean distance between P and the individual v in P^*, when $d(x,P^*)$ represents that between P^* and the individual x in P. d_i is the Euclidean distance between the i-th solution and the (i + 1)-th solution in P, $\overline{d}$ is the average distance, and d_f, d_l respectively represent the Euclidean distance between the two extreme solutions in P and the two endpoints of the real frontier.

4.2. Verification of Improvement Measures Effectiveness

In the process of population evolution, DMOTLBO algorithm uses multiple crossover operators based on probability to generate offspring individuals, which can enhance the global search performance of the current algorithm. At the same time, the introduction of variable neighborhood descent search can improve the local search ability of DMOTLBO algorithm and further improve the quality of solution. The effectiveness of the above methods can be verified by case studies. In this paper, each test example is solved independently 20 times by different algorithms, and statistical analysis is carried out based on performance evaluation indexes. Mean represents the average value of indicators, and Std represents the corresponding standard deviation. Firstly, the DMOTLBO algorithm is used to solve the problem, and the indicators are calculated. Then, the DMOTLBO algorithm without variable neighborhood search is used as algorithm TLBO 1. Finally, the same cases are solved by the TLBO algorithm, which only includes teaching preview operation, teacher training operation, teacher teaching operation and students' mutual learning operation and post-learning review operation, and its average optimization effect is analyzed as the comparative algorithm TLBO 2. The results are shown in Figure 8 as below. Overall, in the vast majority of test examples, DMOTLBO has achieved relatively small IGD, GD and SP values, and its convergence, distribution uniformity and diversity are excellent, which indicates that the mixed use of multiple crossover algorithms and the embedding of variable neighborhood descent search have obviously enhanced the optimization ability of TLBO algorithm and promoted the performance of the algorithm.

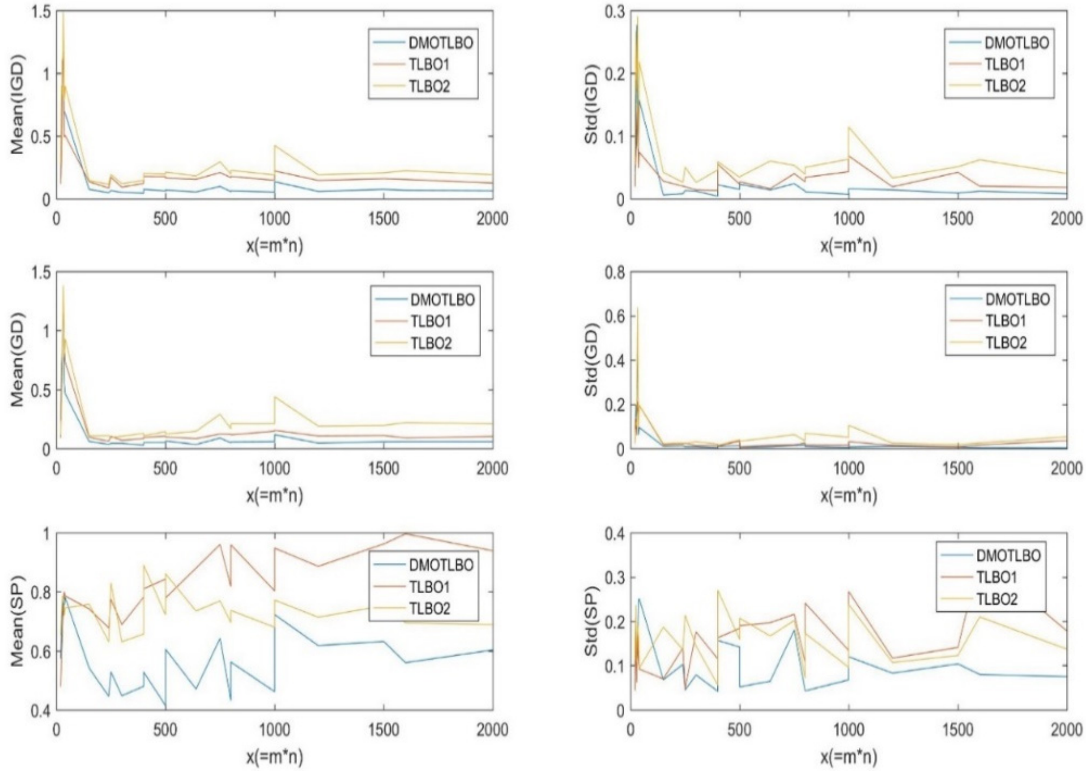

Figure 8. Result of indicators corresponding to the three algorithms DMOTLBO and TLBO1&2.

4.3. Verification of Algorithm Effectiveness

In order to verify the efficiency of DMOTLBO, it is compared with pre-P MOEA/D [18] and HMOMBO [19]. Pre-P MOEA/D is also a mixed multi-objective optimization algorithm based on decomposition, proposing to divide the population before breeding offspring. Meanwhile, HMOMBO is a new multi-objective evolutionary algorithm based on mixed swarm intelligence, integrating monarch butterfly optimization framework and mutating infeasible solutions based on constraints. For each test example, calculate the mean and standard deviation of IGD of the three test algorithms, respectively, and the results are shown in Table 1 where the optimal ones are shown in bold black. The results of GD and SP refer to Tables A1 and A2, respectively, in Appendix A. In order to ensure the fairness of algorithm evaluation, under the level of significance a = 0.05, a t-test is performed on each algorithm according to the method given in Reference [20], and "+" "−" "≈" indicate that the algorithm is superior to, inferior to and similar to the comparison algorithm, respectively.

It can be seen from Table 1 that for the comprehensive indicator IGD, DMOTLBO achieved a winning rate of 18/24 in 24 groups of examples, when the IGD corresponding to pre-P MOEA/D and HMOMBO achieved minimum values for three times, respectively. As far as the convergence indicator GD is concerned, the DMOTLBO algorithm performs better in test examples with the ratio of 20/24, and the numbers corresponding to pre-P MOEA/D and HMOMBO are 2 and 2, respectively. For the distribution indicator SP, the DMOTLBO algorithm obtains the best value in 19 of 24 test cases, and the winning ratios of pre-P MOEA/D and HMOMBO are 2/24 and 3/24, respectively. Generally speaking, the Pareto solution set found by DMOMTLBO algorithm with more competitive average quality has better convergence and distribution compared with the optimization results of the other algorithms. In most cases, it can provide better solutions than the comparison algorithms.

Table 1. Case result of Mean and Std of IGD corresponding to three test algorithms.

NO.	$m \times n$	Pre-P MOEA/D		HMOMBO		DMOTLBO	
		Mean	Std	Mean	Std	Mean	Std
1	2×10	**1.190×10^{-1} (+)**	**2.14×10^{-2}**	2.634×10^{-1} (−)	5.22×10^{-2}	1.812×10^{-1} (−)	4.47×10^{-2}
2	2×12	**3.487×10^{-1} (+)**	**6.96×10^{-2}**	8.924×10^{-1} (−)	3.29×10^{-2}	4.872×10^{-1} (−)	8.30×10^{-2}
3	2×14	9.874×10^{-1} (−)	8.37×10^{-2}	**7.457×10^{-1} (+)**	**1.71×10^{-2}**	9.455×10^{-1} (−)	5.37×10^{-2}
4	2×16	1.198×10 (≈)	1.52×10^{-1}	1.101×10 (≈)	2.53×10^{-1}	**1.043×10 (+)**	**1.51×10^{-1}**
5	2×18	**4.859×10^{-1} (+)**	**5.40×10^{-2}**	6.284×10^{-1} (−)	8.06×10^{-2}	5.872×10^{-1} (≈)	6.89×10^{-2}
6	2×20	6.356×10^{-1} (≈)	7.43×10^{-2}	**6.243×10^{-1} (+)**	**6.51×10^{-2}**	6.291×10^{-1} (≈)	1.61×10^{-1}
7	5×30	1.149×10^{-1} (−)	2.98×10^{-2}	**8.346×10^{-2} (+)**	**1.35×10^{-2}**	9.348×10^{-2} (≈)	2.16×10^{-2}
8	5×50	1.698×10^{-1} (−)	2.04×10^{-2}	1.896×10^{-1} (−)	4.93×10^{-2}	**6.391×10^{-2} (+)**	**1.65×10^{-2}**
9	5×80	1.814×10^{-1} (−)	5.38×10^{-2}	2.031×10^{-1} (−)	5.50×10^{-2}	**7.542×10^{-2} (+)**	**2.22×10^{-2}**
10	5×100	1.673×10^{-1} (−)	2.72×10^{-2}	2.156×10^{-1} (−)	3.55×10^{-2}	**7.092×10^{-2} (+)**	**2.11×10^{-2}**
11	5×150	1.985×10^{-1} (−)	3.89×10^{-2}	3.004×10^{-1} (−)	5.15×10^{-2}	**1.005×10^{-1} (+)**	**2.50×10^{-2}**
12	5×200	2.005×10^{-1} (−)	7.04×10^{-2}	5.023×10^{-1} (−)	9.71×10^{-2}	**1.333×10^{-1} (+)**	**1.54×10^{-2}**
13	8×30	8.587×10^{-2} (−)	2.51×10^{-2}	1.198×10^{-1} (−)	2.60×10^{-2}	**4.479×10^{-2} (+)**	**8.47×10^{-3}**
14	8×50	1.220×10^{-1} (−)	1.30×10^{-2}	1.598×10^{-1} (−)	4.45×10^{-2}	**4.792×10^{-2} (+)**	**3.73×10^{-3}**
15	8×80	1.542×10^{-1} (−)	1.60×10^{-2}	1.914×10^{-1} (−)	5.98×10^{-2}	**5.198×10^{-2} (+)**	**1.46×10^{-2}**
16	8×100	1.812×10^{-1} (−)	3.40×10^{-2}	2.306×10^{-1} (−)	4.84×10^{-2}	**6.582×10^{-2} (+)**	**9.46×10^{-3}**
17	8×150	1.630×10^{-1} (−)	2.37×10^{-2}	2.012×10^{-1} (−)	3.28×10^{-2}	**5.872×10^{-2} (+)**	**1.50×10^{-2}**
18	8×200	1.562×10^{-1} (−)	2.10×10^{-2}	2.357×10^{-1} (−)	6.20×10^{-2}	**6.727×10^{-2} (+)**	**1.29×10^{-2}**
19	10×30	9.304×10^{-2} (−)	1.41×10^{-2}	1.338×10^{-1} (−)	2.58×10^{-2}	**4.824×10^{-2} (+)**	**1.24×10^{-2}**
20	10×50	1.756×10^{-1} (−)	2.25×10^{-2}	2.005×10^{-1} (−)	3.20×10^{-2}	**6.383×10^{-2} (+)**	**1.49×10^{-2}**
21	10×80	1.753×10^{-1} (−)	2.75×10^{-2}	1.987×10^{-1} (−)	4.14×10^{-2}	**5.498×10^{-2} (+)**	**1.31×10^{-2}**
22	10×100	1.527×10^{-1} (−)	4.27×10^{-2}	2.142×10^{-1} (−)	6.30×10^{-2}	**5.340×10^{-2} (+)**	**7.36×10^{-3}**
23	10×150	1.732×10^{-1} (−)	4.35×10^{-2}	2.278×10^{-1} (−)	5.20×10^{-2}	**7.502×10^{-2} (+)**	**1.04×10^{-2}**
24	10×200	1.368×10^{-1} (−)	1.83×10^{-2}	2.012×10^{-1} (−)	4.31×10^{-2}	**6.109×10^{-2} (+)**	**8.48×10^{-3}**
+/−/≈		3/19/2		3/20/1		18/3/3	

4.4. Cases Analysis

In order to further verify the effectiveness and efficiency of the model and algorithm proposed in this paper, simulation cases are given here. A rectangular coordinate system XOY is set up due to the fact that the related programming problems are considered in a two-dimensional environment. The task area is set to a rectangular area of 60 km × 60 km, and the emergency rescue flight base are set to the point (30, 30). The flight performance of the UAV is shown in Table 2. Randomly generate a corresponding number of task target locations in the simulation area. On this basis, the number of rescue UAVs m is set to 3, the number of target points n is set to 30, that is, 3 × 30 examples are generated, and then 4 × 40 and 5 × 50 examples are constructed, respectively. The examples select the UAVs from Table 2 in order from top to bottom, respectively. Other relevant parameter settings are shown in Table A3 in Appendix A, including advance penalty cost coefficient F_n, delay penalty cost coefficient D_n and expected delivery time of materials E_n. Keep the scenarios and parameters in this example the same, then solve the three groups of cases with DMOTLBO, pre-P MOEA/D and HMOMBO algorithms mentioned above.

Table 2. The flight performance of the UAVs chosen.

No.	UAV No.	Brand Model	Average Cruise Speed (km/h)	Max-Endurance (h)	Maximum Load Capacity (kg)
1	M1	Zongheng CW100	100	10	20
2	M2	Zongheng CW30	90	6	6
3	M3	Zongheng CW10	81	1.6	3
4	M4	Ebee a	80	1.6	2
5	M5	Ebee b	70	0.9	1

Take the 5 × 50 example as an example, there are some conflicts between the two optimization objectives of the current model, and Figure 9 shows an optimal solution obtained by DMOTLBO algorithm. The planning result of DMOTLBO is: M1 flying orange task flow and line, M2 flying red task flow and line, M3 flying gray task flow and line, M4 flying blue task flow and line, and M5 flying green task flow and line, as shown in the table below.

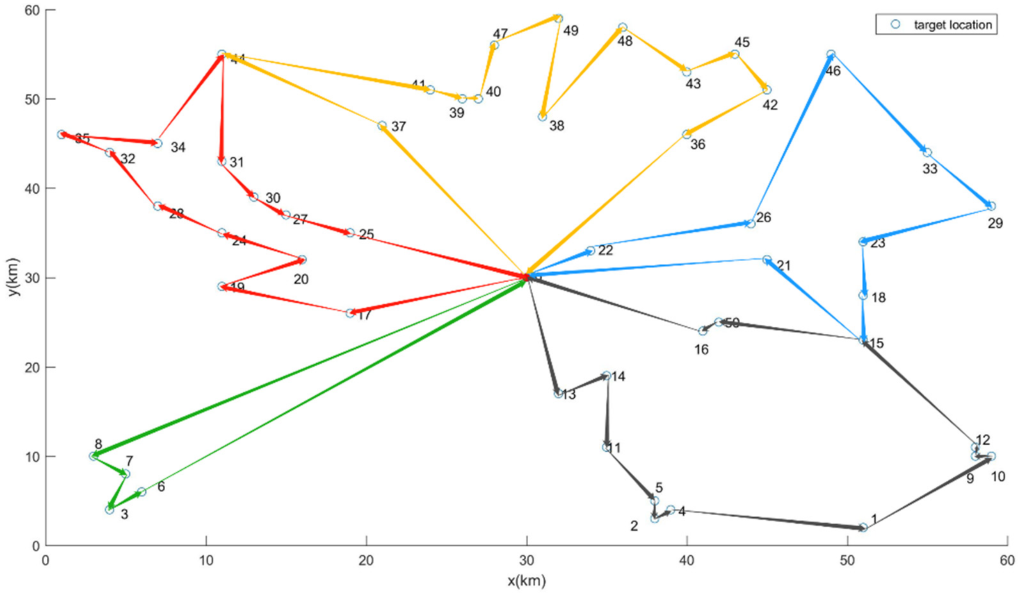

Figure 9. Sketch map of a planning result of the DMOTLBO algorithm.

Additionally, UAVs are planned to perform emergency logistics tasks in sequence. The outcome data of the planning result is shown in Table 3.

Table 3. The outcome data of the planning result of the DMOTLBO algorithm.

No.	UAV No.	Route Color	Route	Total Length/km	Last Task Flight Length/km	Duration/h	Total Material Quantity/kg	Constraint Satisfaction
1	M1	Orange	0-37-44-41-39-40-47-49-38-48-43-45-42-36-0	122.608	103.740	1.037	19.732	Satisfy
2	M2	Red	0-17-19-20-24-28-32-35-34-44-31-30-27-25-0	99.933	87.850	0.976	5.910	Satisfy
3	M3	Gray	0-13-14-11-5-2-4-1-10-9-12-15-50-16-0	97.416	84.887	1.048	2.850	Satisfy
4	M4	Blue	0-22-26-46-33-29-23-18-15-21-0	100.721	85.589	1.070	1.960	Satisfy
5	M5	Green	0-8-7-3-6-0	78.263	41.494	0.593	0.928	Satisfy

Then, calculate the same problems with the pre-P MOEA/D and HMOMBO algorithms mentioned above. Table 4 shows the mean and standard deviation of IGD, GD and SP of the test results obtained by the three algorithms. The data show that, compared with the pre-P MOEA/D algorithm and HMOMBO algorithm, the indicators of the test results obtained by the DMOTLBO algorithm are better, indicating that its non-inferior solution set is better in convergence and distribution, which effectively verifies the practicability and efficiency of the current algorithm for UAV emergency logistics task planning.

Table 4. Data comparison of Mean and Std of IGD, GD, and SP, corresponding to DMOTLBO and the two contrast algorithms, respectively.

Case	Indicator	DMOTLBO		Pre-P MOEA/D		HMOMBO	
		Mean	Std	Mean	Std	Mean	Std
3 × 30	IGD	$\mathbf{5.63 \times 10^{-2}}$	$\mathbf{8.90 \times 10^{-3}}$	1.07×10^{-1}	2.18×10^{-2}	8.21×10^{-2}	2.01×10^{-2}
	GD	$\mathbf{2.16 \times 10^{-2}}$	$\mathbf{4.50 \times 10^{-3}}$	9.50×10^{-2}	1.68×10^{-2}	1.38×10^{-1}	4.79×10^{-2}
	SP	$\mathbf{4.31 \times 10^{-1}}$	$\mathbf{4.05 \times 10^{-2}}$	7.28×10^{-1}	9.13×10^{-2}	7.49×10^{-1}	1.41×10^{-1}
4 × 40	IGD	$\mathbf{4.75 \times 10^{-2}}$	$\mathbf{6.57 \times 10^{-3}}$	1.48×10^{-1}	1.96×10^{-2}	1.56×10^{-1}	3.87×10^{-2}
	GD	$\mathbf{3.85 \times 10^{-2}}$	$\mathbf{1.03 \times 10^{-2}}$	1.16×10^{-1}	1.85×10^{-2}	1.26×10^{-1}	5.21×10^{-2}
	SP	$\mathbf{3.77 \times 10^{-1}}$	$\mathbf{9.83 \times 10^{-2}}$	7.79×10^{-1}	1.09×10^{-1}	8.53×10^{-1}	2.15×10^{-1}
5 × 50	IGD	$\mathbf{6.67 \times 10^{-2}}$	$\mathbf{1.56 \times 10^{-2}}$	1.71×10^{-1}	1.87×10^{-2}	1.93×10^{-1}	5.04×10^{-2}
	GD	$\mathbf{5.88 \times 10^{-2}}$	$\mathbf{7.98 \times 10^{-3}}$	1.29×10^{-1}	2.39×10^{-2}	1.31×10^{-1}	2.57×10^{-2}
	SP	$\mathbf{5.42 \times 10^{-1}}$	$\mathbf{4.96 \times 10^{-2}}$	7.72×10^{-1}	5.62×10^{-2}	8.47×10^{-1}	2.07×10^{-1}

To sum up, relative to existing algorithms, the performance advantages of the DMOTLBO algorithm provides a new idea for the existing research, which benefit from the following five aspects: (1) adopting a multi-objective algorithm framework based on the decomposition mechanism, combining with an aggregation function, and using a uniform weight design of mixture can make the solution distribution better; (2) combined with the TLBO algorithm to solve the sub-problems, the algorithm has the advantages of no parameters and high efficiency in solving optimization problems; (3) using the sequence coding method, it can directly perform the global search based on the mechanism of the standard TLBO algorithm in the solution space of discrete problems, thus obviously improving the global search efficiency of the original algorithm; (4) using a variety of crossover mutation operators based on probability to further improve the optimization efficiency; (5) the local search ability of the algorithm is ensured by embedding variable neighborhood descent search to search carefully near the high-quality solution.

5. Conclusions

In this paper, aiming at the task planning of UAV emergency material delivery, a mathematical optimization model is established with the goal of minimizing the task completion time and the penalty cost of advance/delay, and a discrete multi-objective teaching–learning-based optimization (DMOTLBO) algorithm is proposed. The Pareto frontier approximation problem is transformed into a set of single objective sub-problems by the decomposition mechanism of the algorithm, and each sub-problem is solved by the improved discrete TLBO algorithm. According to the characteristics of the problem, TLBO algorithm is improved by discretization, and an individual update method is constructed based on the probability fusion of various mutation evolution operators. At the same time, variable neighborhood descent search is introduced to enhance the local search ability. Based on the multi-level comparative experiment, the improvement measures and effectiveness of DMOTLBO algorithm are verified. Finally, combined with specific case analysis, the practicability and efficiency of DMOTLBO algorithm in solving the multi-objective emergency logistics task planning problem of multiple unmanned aerial vehicles are further verified. The key innovations and merits of the UAV planning method proposed lie in the fact that the TLBO algorithm being combined with the decomposition mechanism is introduced for the first time to solve the UAV mission planning problem, and the algorithm with outstanding search capability and efficiency is improved by discretization and descent search. The proposed method provides a new idea for UAV mission planning research and fills the blank of multi-target mission planning of rescue UAV in emergency logistics to some extent. Nevertheless, the UAV task planning method is only considered in the two-dimensional environment, and the UAV is simplified at the same time, ignoring the UAV dynamic performance, wind and other factors, as well as complex situations such as the mid-mission change, etc. In the next step, further research will be carried out and higher dimensions and more influencing factors will be considered to improve the practicability of the method.

Author Contributions: Conceptualization, M.H.; methodology, H.Z.; software, L.Z.; data curation, L.Z.; writing—original draft preparation, M.T.; writing—review and editing, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant No. 71971114, 52002178) and Natural Science Foundation of Jiangsu Province (Grant No. BK20190416).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge with thanks the support for this work by the College of Civil Aviation at Nanjing University of Aeronautics and Astronautics.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Case result of Mean and Std of GD corresponding to three test algorithms.

NO.	$m \times n$	Pre-P MOEA/D		HMOMBO		DMOTLBO	
		Mean	Std	Mean	Std	Mean	Std
1	2 × 10	**8.781×10^{-2} (+)**	**4.15×10^{-2}**	1.234×10^{-1} (+)	5.89×10^{-2}	3.882×10^{-1}(−)	5.69×10^{-2}
2	2 × 12	5.349×10^{-1} (+)	1.96×10^{-1}	**4.324×10^{-1} (+)**	**1.13×10^{-1}**	8.722×10^{-1} (−)	3.07×10^{-1}
3	2 × 14	4.871×10^{-1} (+)	9.86×10^{-2}	**2.048×10^{-1} (+)**	**5.98×10^{-2}**	6.482×10^{-1} (−)	1.19×10^{-1}
4	2 × 16	3.345×10^{-1} (−)	2.11×10^{-1}	3.651×10^{-1} (−)	1.64×10	**2.934×10^{-1} (+)**	**1.51×10^{-1}**
5	2 × 18	8.287×10^{-1} (−)	2.18×10^{-1}	6.787×10^{-1} (−)	1.35×10^{-1}	**5.652×10^{-1} (+)**	**9.46×10^{-2}**
6	2 × 20	7.083×10^{-1} (−)	1.89×10^{-1}	5.128×10^{-1} (−)	1.96×10^{-1}	**4.378×10^{-1} (+)**	**8.99×10^{-2}**
7	5 × 30	**9.980×10^{-2} (≈)**	**1.82×10^{-2}**	5.810×10^{-1} (−)	3.01×10^{-2}	1.004×10^{-1} (≈)	5.87×10^{-2}
8	5 × 50	9.650×10^{-2} (−)	2.27×10^{-2}	9.478×10^{-2} (−)	2.10×10^{-2}	**4.598×10^{-2} (+)**	**7.29×10^{-3}**
9	5 × 80	1.021×10^{-1} (−)	1.38×10^{-2}	9.842×10^{-2} (−)	1.28×10^{-2}	**6.041×10^{-2} (+)**	**1.01×10^{-2}**
10	5 × 100	9.987×10^{-2} (−)	7.12×10^{-3}	1.205×10^{-1} (−)	3.46×10^{-2}	**7.032×10^{-2} (+)**	**5.87×10^{-3}**
11	5 × 150	1.342×10^{-1} (−)	1.64×10^{-2}	2.413×10^{-1} (−)	6.27×10^{-2}	**8.593×10^{-2} (+)**	**1.45×10^{-2}**
12	5 × 200	1.542×10^{-1} (−)	6.89×10^{-2}	4.287×10^{-1} (−)	1.12×10^{-1}	**1.482×10^{-1} (+)**	**3.81×10^{-2}**
13	8 × 30	6.583×10^{-2} (−)	1.97×10^{-2}	9.840×10^{-2} (−)	2.05×10^{-2}	**3.578×10^{-2} (+)**	**1.28×10^{-2}**
14	8 × 50	7.333×10^{-2} (−)	1.49×10^{-2}	1.048×10^{-1} (−)	2.05×10^{-2}	**3.135×10^{-2} (+)**	**9.12×10^{-3}**
15	8 × 80	7.164×10^{-2} (−)	1.53×10^{-2}	1.258×10^{-1} (−)	4.70×10^{-2}	**3.359×10^{-2} (+)**	**6.98×10^{-3}**
16	8 × 100	1.026×10^{-1} (−)	1.53×10^{-2}	1.893×10^{-1} (−)	6.47×10^{-2}	**5.293×10^{-2} (+)**	**8.79×10^{-3}**
17	8 × 150	1.187×10^{-1} (−)	1.66×10^{-2}	1.839×10^{-1} (−)	3.11×10^{-2}	**4.489×10^{-2} (+)**	**1.30×10^{-2}**
18	8 × 200	9.349×10^{-2} (−)	2.05×10^{-2}	2.123×10^{-1} (−)	2.76×10^{-2}	**4.872×10^{-2} (+)**	**4.64×10^{-3}**
19	10 × 30	7.142×10^{-2} (−)	1.49×10^{-2}	9.754×10^{-2} (−)	3.24×10^{-2}	**5.475×10^{-2} (+)**	**9.58×10^{-3}**
20	10 × 50	1.359×10^{-1} (−)	3.49×10^{-2}	1.542×10^{-1} (−)	4.12×10^{-2}	**5.872×10^{-2} (+)**	**1.49×10^{-2}**
21	10 × 80	1.135×10^{-1} (−)	2.43×10^{-2}	1.672×10^{-1} (−)	4.45×10^{-2}	**5.342×10^{-2} (+)**	**2.13×10^{-2}**
22	10 × 100	1.374×10^{-1} (−)	1.53×10^{-2}	1.983×10^{-1} (−)	4.88×10^{-2}	**6.234×10^{-2} (+)**	**5.23×10^{-3}**
23	10 × 150	1.203×10^{-1} (−)	9.15×10^{-3}	2.012×10^{-1} (−)	1.81×10^{-2}	**5.634×10^{-2} (+)**	**7.35×10^{-3}**
24	10 × 200	9.870×10^{-2} (−)	3.13×10^{-2}	1.987×10^{-1} (−)	5.13×10^{-2}	**6.129×10^{-2} (+)**	**3.87×10^{-3}**
+/−/≈		3/20/1		3/21/0		20/3/1	

Table A2. Case result of Mean and Std of SP corresponding to three test algorithms.

NO.	$m \times n$	Pre-P MOEA/D		HMOMBO		DMOTLBO	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
1	2×10	**5.098×10^{-1} (+)**	**4.68×10^{-2}**	5.892×10^{-1} (−)	1.09×10^{-1}	5.242×10^{-1} (≈)	7.21×10^{-2}
2	2×12	**6.102×10^{-1} (+)**	**1.82×10^{-1}**	7.069×10^{-1} (−)	2.13×10^{-1}	6.331×10^{-1} (≈)	1.90×10^{-1}
3	2×14	7.013×10^{-1} (≈)	1.38×10^{-1}	**6.213×10^{-1} (+)**	**8.99×10^{-2}**	7.065×10^{-1} (−)	9.01×10^{-2}
4	2×16	7.315×10^{-1} (≈)	1.82×10^{-1}	7.637×10^{-1} (−)	1.92×10^{-1}	**7.103×10^{-1} (+)**	**1.19×10^{-1}**
5	2×18	7.896×10^{-1} (−)	1.59×10^{-1}	**6.309×10^{-1} (+)**	**1.36×10^{-1}**	6.498×10^{-1} (≈)	2.39×10^{-1}
6	2×20	7.763×10^{-1} (−)	9.44×10^{-2}	**6.551×10^{-1} (+)**	**9.21×10^{-2}**	6.695×10^{-1} (≈)	1.93×10^{-1}
7	5×30	7.121×10^{-1} (−)	8.93×10^{-2}	7.380×10^{-1} (−)	1.30×10^{-1}	**5.392×10^{-1} (+)**	**6.57×10^{-2}**
8	5×50	6.952×10^{-1} (−)	5.03×10^{-2}	7.953×10^{-1} (−)	2.10×10^{-1}	**5.176×10^{-1} (+)**	**5.01×10^{-2}**
9	5×80	8.245×10^{-1} (−)	1.61×10^{-1}	9.012×10^{-1} (−)	1.98×10^{-1}	**5.109×10^{-1} (+)**	**1.55×10^{-1}**
10	5×100	7.598×10^{-1} (−)	1.84×10^{-1}	8.824×10^{-1} (−)	1.87×10^{-1}	**5.897×10^{-1} (+)**	**5.72×10^{-2}**
11	5×150	9.469×10^{-1} (−)	2.14×10^{-1}	7.583×10^{-1} (−)	2.00×10^{-1}	**6.309×10^{-1} (+)**	**1.66×10^{-1}**
12	5×200	9.281×10^{-1} (−)	2.47×10^{-1}	7.813×10^{-1} (−)	2.50×10^{-1}	**7.031×10^{-1} (+)**	**1.17×10^{-1}**
13	8×30	6.481×10^{-1} (−)	1.68×10^{-1}	6.311×10^{-1} (−)	1.34×10^{-1}	**4.334×10^{-1} (+)**	**9.48×10^{-2}**
14	8×50	7.620×10^{-1} (−)	1.15×10^{-1}	6.775×10^{-1} (−)	5.64×10^{-2}	**4.307×10^{-1} (+)**	**4.72×10^{-2}**
15	8×80	8.678×10^{-1} (−)	2.37×10^{-1}	7.247×10^{-1} (−)	1.80×10^{-1}	**4.456×10^{-1} (+)**	**6.35×10^{-2}**
16	8×100	9.456×10^{-1} (−)	2.31×10^{-1}	7.271×10^{-1} (−)	1.42×10^{-1}	**5.589×10^{-1} (+)**	**5.09×10^{-2}**
17	8×150	8.360×10^{-1} (−)	1.12×10^{-1}	6.946×10^{-1} (−)	1.10×10^{-1}	**5.998×10^{-1} (+)**	**8.35×10^{-2}**
18	8×200	9.568×10^{-1} (−)	3.40×10^{-1}	7.050×10^{-1} (−)	2.10×10^{-1}	**5.375×10^{-1} (+)**	**7.01×10^{-2}**
19	10×30	7.093×10^{-1} (−)	1.79×10^{-1}	6.542×10^{-1} (−)	1.65×10^{-1}	**4.465×10^{-1} (+)**	**7.91×10^{-2}**
20	10×50	8.130×10^{-1} (−)	1.89×10^{-1}	7.129×10^{-1} (−)	1.90×10^{-1}	**3.923×10^{-1} (+)**	**1.87×10^{-1}**
21	10×80	8.287×10^{-1} (−)	1.67×10^{-1}	7.462×10^{-1} (−)	7.02×10^{-2}	**4.102×10^{-1} (+)**	**4.63×10^{-2}**
22	10×100	7.838×10^{-1} (−)	1.22×10^{-1}	7.014×10^{-1} (−)	9.56×10^{-2}	**4.509×10^{-1} (+)**	**8.78×10^{-2}**
23	10×150	9.422×10^{-1} (−)	1.37×10^{-1}	7.463×10^{-1} (−)	1.20×10^{-1}	**6.120×10^{-1} (+)**	**1.00×10^{-1}**
24	10×200	9.092×10^{-1} (−)	1.68×10^{-1}	6.699×10^{-1} (−)	1.40×10^{-1}	**5.873×10^{-1} (+)**	**7.76×10^{-2}**
+/−/≈		2/20/2		3/21/0		19/1/4	

Table A3. Relevant parameter settings in simulation cases.

n	F_n	D_n	E_n	q_n	n	F_n	D_n	E_n	q_n
1	0.5	0.9	20	0.200	26	0.2	0.7	63	0.300
2	0.2	0.8	24	0.150	27	0.2	1.0	50	0.150
3	0.4	0.8	32	0.108	28	0.4	0.9	24	0.310
4	0.3	0.8	48	0.150	29	0.5	0.8	66	0.160
5	0.3	0.9	50	0.200	30	0.3	0.8	51	0.200
6	0.2	0.9	60	0.108	31	0.2	0.7	52	0.200
7	0.2	0.7	55	0.112	32	0.3	0.9	33	0.400
8	0.4	0.7	29	0.600	33	0.3	0.8	37	0.300
9	0.3	0.7	09	0.200	34	0.1	0.9	38	0.200
10	0.2	0.7	27	0.150	35	0.1	0.6	46	0.300
11	0.4	0.6	44	0.250	36	0.4	1.0	66	1.250
12	0.3	0.7	28	0.200	37	0.3	1.0	16	1.200
13	0.2	0.8	05	0.200	38	0.5	0.9	13	1.350
14	0.1	0.7	49	0.200	39	0.3	1.0	05	2.580
15	0.4	0.7	55	1.000	40	0.3	0.8	45	1.362
16	0.3	0.7	67	0.150	41	0.4	0.8	59	1.350
17	0.2	0.8	14	0.200	42	0.4	0.7	40	1.320
18	0.1	0.8	01	0.100	43	0.2	0.6	16	1.732
19	0.4	0.7	31	1.000	44	0.4	1.0	04	4.000
20	0.1	0.6	14	0.300	45	0.5	1.0	63	1.410
21	0.4	0.7	43	0.200	46	0.5	0.9	23	0.100
22	0.4	0.8	14	0.200	47	0.3	0.7	76	1.320
23	0.2	0.8	37	0.100	48	0.1	0.7	59	1.550
24	0.1	0.7	04	0.500	49	0.3	0.7	10	1.308
25	0.4	0.7	44	0.150	50	0.3	0.7	13	0.300

References

1. Zou, P.; Deng, J.; Liu, L.; Ou, F. UAV transportation and flight path planning in geological disaster response. *Geomat. Spat. Inf. Technol.* **2017**, *40*, 15–17, 20.
2. Wang, C.; Wu, L.; Yan, C.; Wang, Z.; Long, H.; Yu, C. Coactive design of explainable agent-based mission planning and deep reinforcement learning for human-UAVs teamwork. *Chin. J. Aeronaut.* **2020**, *33*, 2930–2945. [CrossRef]
3. Kim, K.S.; Kim, H.Y.; Choi, H.L. A bid-based grouping method for communication-efficient decentralized multi-UAV task allocation. *Int. J. Aeronaut. Space Sci.* **2020**, *21*, 290–302. [CrossRef]
4. Siemiatkowska, B.; Stecz, W. A Framework for Planning and Execution of Drone Swarm Missions in a Hostile Environment. *Sensors* **2021**, *21*, 4150. [CrossRef] [PubMed]
5. Fu, Z.; Mao, Y.; He, D.; Yu, J.; Xie, G. Secure multi-uav collaborative task allocation. *IEEE Access* **2019**, *7*, 35579–35587. [CrossRef]
6. Qi, X.; Li, B.; Fan, Y.; Liu, L. Review on mission planning of multi-uav under multi-constraints. *J. Intell. Syst.* **2020**, *15*, 204–217.
7. Ye, F.; Chen, J.; Tian, Y.; Jiang, T. Cooperative Multiple Task Assignment of Heterogeneous UAVs Using a Modified Genetic Algorithm with Multi-type-gene Chromosome Encoding Strategy. *J. Intell. Robot. Syst.* **2020**, *100*, 615–627. [CrossRef]
8. Yan, M.; Yuan, H.; Xu, J.; Yu, Y.; Jin, L. Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm. *EURASIP J. Adv. Signal Process.* **2021**, *2021*, 94. [CrossRef]
9. Wu, L.; Cui, Y.; Zhu, H.; Cui, J. Research on the multi-rotor UAV multi-task assignment based on discrete particle swarm optimization algorithm. *Fire Sci. Technol.* **2020**, *39*, 662.
10. Liu, Y.; Zhang, X.; Zhang, Y.; Guan, X. Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach. *Chin. J. Aeronaut.* **2019**, *32*, 1504–1519. [CrossRef]
11. Tan, W.; Hu, Y.; Zhao, Y.; Li, W.; Li, Y.; Zhang, X. Heterogeneous Multi UAV Mission Planning Based on Ant Colony Algorithm Powered BP Neural Network. *Comput. Intell. Neurosci.* **2021**, *32*, 382–396. [CrossRef] [PubMed]
12. Yan, F. Gauss interference ant colony algorithm-based optimization of UAV mission planning. *J. Supercomput.* **2020**, *76*, 1170–1179. [CrossRef]
13. Tanabe, R.; Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Appl. Soft Comput.* **2020**, *89*, 106078. [CrossRef]
14. Li, M.; Lei, D. Novel imperialist competitive algorithm for any-objective flexible job shop scheduling. *Control. Theory Appl.* **2019**, *36*, 893–901.
15. Li, Y.H.; Yong, L.Q.; Tuo, S.H. Teaching and learning optimization algorithm based on improved crossover-self-study strategy. *Trans. Intell. Syst.* **2021**, *16*, 313–322.
16. Li, M. Research on UAV mission Planning Method Based on Intelligent Optimization and RRT Algorithm. Ph.D. Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2012.
17. Lin, S.W.; Ying, K.C. A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems. *Int. J. Prod. Res.* **2015**, *53*, 1065–1076. [CrossRef]
18. Guo, Y. Multi-Objective Evolutionary Algorithm and Its Research on Scheduling Problem. Master's Thesis, Hunan University, Changsha, China, 2019.
19. Rui, H. Research on Multi-Objective Optimization Algorithm Based on Hybrid Swarm Intelligence. Master's Thesis, Dalian Maritime University, Dalian, China, 2020.
20. Zhang, C.J.; Li, X.Y.; Gao, L.; Wu, Q. An improved electromagnetism-like mechanism algorithm for constrained optimization. *Expert Syst. Appl.* **2013**, *40*, 5621–5634. [CrossRef]