This work presents a comparison of three “state-of-the-art” multi-objective evolutionary algorithms, the non-dominated sorting genetic algorithm II (NSGA-II), the decomposition-based multi-objective evolutionary algorithm with the $\varepsilon$-constraint framework (DMOEA-$\varepsilon$C) and decomposition-based multi-objective evolutionary algorithm with the adaptive weight vector adjustment (MOEA/D-AWA), when applied to the task planning of unmanned aerial vehicles (UAVs) with heterogeneous payloads. The problem studied in this paper is a mixed-variable optimization problem, which involves the task allocation with inequality constraints and the path planning with neighborhood and curvature constraints. To solve this problem, a bi-level solution scheme is proposed. The first level, the task allocation is solved by the “state-of-the-art” algorithm with the proposed encoding, the crossover operator, and a series of mutation operators. In the second level, the path planning is solved by a sampling-based heuristic method. In order to verify the performance of these “state-of-the-art” algorithms on the problem of this paper, several random simulation instances with different scales are constructed, and the calculation results are analyzed by using multiple performance evaluation index. The comparative results show that DMOEA-$\varepsilon$C can find significantly better Pareto fronts in most instances.

Keywords: multi-objective evolutionary algorithm, UAVs, task planning, task allocation, path planning

1. Introduction

Compared with the single unmanned aerial vehicle (UAV), multi-UAV system has the advantages of high speed, reliability and flexibility for cooperation [1, 2]. Complex tasks, such as wide area search, intelligence surveillance and reconnaissance, require multiple UAVs to perform in a cooperative way. Moreover, multiple unmanned aerial systems are gradually replacing single unmanned aerial vehicles in performing various complex tasks [3]. However, multiple UAVs working together increases the scale of the problem, which increases the computational complexity exponentially. In addition, the cooperation of multi-UAV is also affected by other factors, such as the payload heterogeneity, cooperation constraints, and conflicts between UAVs, etc [4, 5]. Therefore, multiple UAVs cooperative task planning is a challenging problem.

The assignment for the multi-UAV and multi-target and the corresponding path planning play two important components of the task planning system. Many scholars devote themselves to the research of those problems. Zhao et al. proposed a unified model for the assignment of multi-UAV and multi-target, and proposed an improved discrete mapping differential evolution to solve the assignment problem [4]. Different from the general assignment problem for the multi-UAV and multi-task, Guo et al. studied the coalition formation (CF) problem which is an important problem focusing on allocating agents to different tasks. In this study, the inequality constraint between multiple UAVs with heterogeneous capabilities and tasks with different requirements is considered. The authors proposed an efficient genetic algorithm with heuristic initialization and a repair strategy (GAHIR) to solve the CF problem [5]. Zhang et al. studied the curvature-constrained task planning for multi-UAV which is regarded as a multi-vehicle Dubins traveling salesman problem with neighborhood (m-DTSPN), and proposed a memetic algorithm to solve it [6].

The above research mainly considers the optimization of a single objective, such as the resource consumption or the makespan. Since it is necessary to balance multiple objectives in actual tasks, many researchers also consider the multi-objective optimization for those problems. In the process of solving such problems, various multi-objective evolutionary algorithms (MOEA), which are widely used for approximating the true Pareto front (PF) of the multi-objective optimization problem (MOP). Ines et al. studied the UAVs trajectory optimization for data pickup and delivery optimization with a time window. Three optimization objectives are considered, including the flying distance, the schedule duration, and the number of the assigned UAVs. The authors also explained...
how to use the non-dominated sorting genetic algorithm II (NSGA-II) to solve this problem [7]. Zhen et al. studied the rotary unmanned aerial vehicles path planning in the rough terrain. Three performance criterions, including the flight height, flight length and turning angle, are considered, and an improved multi-objective particle swarm optimization algorithm (MOPSO) was developed [8].

After some researchers used MOEAs to solve the task assignment and path planning problems, we also used the MOEAs framework to solve the task planning problem of UAVs with heterogeneous payloads, aiming to generate more diversified and richer solutions. In addition, this study combines the characteristics of the literature [4–6], and makes the following innovations and contributions:

(1) This paper studies the multi-objective task planning of UAVs with heterogeneous payloads. This problem includes both task allocation and path planning, which is a mixed-variable optimization problem. To solve this problem, a bi-level solution scheme is proposed. In the first level, the task allocation is solved by the state-of-the-art algorithm, e.g. NSGA-II, the ε-constraint framework (DMOEA-εC), and the decomposition-based multi-objective evolutionary algorithm with adaptive weight vector adjustment (MOEAs/D-AWA). In the second level, as a subproblem of the task allocation, path planning is efficiently solved by a proposed sampling-based heuristic method.

(2) According to the characteristics of the problem, a new encoding is proposed to represent the pairing of UAVs and tasks. In addition, the crossover operator and a set of mutation operators are designed.

(3) Performance of NSGA-II, DMOEA-εC, and MOEAs/D-AWA on the proposed problem are assessed and analysed based on multiple performance evaluation indexes.

This paper is organized as follows. Section 2 presents the formulation of the multi-objective task planning of UAVs with heterogeneous payloads. Section 3 gives a detailed description of the problem solving. Section 4 evaluates different algorithms through a series of computational experiments and analyses. Section 5 concludes the paper.

2. Problem formulation

The multi-objective task planning of UAVs with heterogeneous payloads studied in this paper considers both the task allocation and the path planning, in which the path planning is regarded as a subproblem of the task allocation. Figure 1 shows the pairing results of UAVs with different payloads and tasks with different requirements, as well as the path planning results of each group of UAVs. It can be seen that UAVs and tasks are divided into orange, green and blue groups, and each group of UAVs is competent for its corresponding tasks. In the path planning, UAVs fly to the assigned targets in a leader-following mode. It is noted that the grey UAV shown in Fig. 1 represents the unassigned ones.

For this problem, two optimization objectives, i.e., the number of the assigned UAVs and the makespan, are considered, and the mathematical modeling is shown as follows.

2.1 Task allocation

Suppose that there are a set of n UAVs with heterogeneous payloads, and m tasks with different requirements. The set of UAVs and tasks are denoted as Eq. (1) and Eq. (3).

\[ U = \{u_1, u_2, \cdots, u_n\} \]  \hspace{1cm} (1)

where \( u_i \) (\( i \in \{1, 2, \cdots, n\} \)) represents the \( i \)th UAV. \( u_i \) may carry \( k \) kinds of payloads, and the payloads carried by \( u_i \) can be denoted by the vector \( B_i \). \( b_i^j = 1 \) / 0 represents whether \( u_i \) is equipped with the \( j \)th payload or not.

\[ B_i = [b_i^1, b_i^2, \cdots, b_i^k] \]  \hspace{1cm} (2)

\[ T = \{t_1, t_2, \cdots, t_m\} \]  \hspace{1cm} (3)

where \( t_i \) (\( i \in \{1, 2, \cdots, m\} \)) represents the \( i \)th task. Each \( t_i \) has different requirements for different payloads, and the requirement vector of detection success rate is represented by \( Q_{t_i} \).

\[ Q_{t_i} = [q_{t_i}^1, q_{t_i}^2, \cdots, q_{t_i}^k] \]  \hspace{1cm} (4)

where \( q_{t_i}^j \) (\( j \in \{1, 2, \cdots, k\} \)) represents the requirement of \( t_i \) on the detection success rate of the \( j \)th payload.

It should be noted that the detection success rate of a same payload for different tasks will be different due to the different environments and characteristics of tasks. The detection success rate of different payloads on different tasks can be expressed as \( P_{2T} \).

\[
P_{2T} = \begin{bmatrix}
d_{11} & d_{12} & \cdots & d_{1k} \\
d_{21} & d_{22} & \cdots & d_{2k} \\
\vdots & \vdots & & \vdots \\
d_{m1} & d_{m2} & \cdots & d_{mk}
\end{bmatrix}
\]  \hspace{1cm} (5)
Then, to represent the grouping and pairing between tasks and UAVs, the decision vectors are defined as follows:

\[ G = [r_1, r_2, \ldots, r_m, s_1, s_2, \ldots, s_{m-1}] \] \hspace{1cm} (6)

\[ A = [a_1, a_2, \ldots, a_n] \] \hspace{1cm} (7)

where \( G \) represents the grouping of tasks. \( [r_1, r_2, \ldots, r_m, s_1, s_2, \ldots, s_{m-1}] \) represents a series of tasks with random sequence, and \( [s_1, s_2, \ldots, s_{m-1}] \) is a separation vector, \( s_j = 1 / 0 \) indicates whether \( t_i \) and \( t_{i+1} \) are separated or not, \( i \in \{1,2,\ldots,m-1\} \). \( A \) represents the assignment of UAVs, and \( a_i \in \{0,1,2,\ldots,m\} \), \( i \in \{1,2,\ldots,n\} \). \( a_i = 0 \) means that no task is assigned to \( a_i \).

Equations (6) and (7) reflect the grouping and pairing between tasks and UAVs. For each group of UAVs, the total ability of the group is not lower than the requirement of the task in any ability type. Suppose that there are \( n \) UAVs in a certain UAV group, and \( m \) tasks are assigned to the UAV group. For the task \( t_i \) (\( i \in \{1,2,\ldots,m\} \)), the total ability of the UAV group on the \( j \)th payload (\( j \in \{1,2,\ldots,k\} \)) should satisfy the following constraint:

\[ 1 - (1 - b_1^i * d_{s1}^j) * (1 - b_2^i * d_{s2}^j) * \cdots * (1 - b_n^i * d_{sn}^j) \geq q_0 \] \hspace{1cm} (8)

If all UAV groups are competent for all the tasks assigned to them, the first objective function, the number of the assigned UAVs, can be calculated by the following equations.

\[ y = [y_1, y_2, \ldots, y_n] \] \hspace{1cm} (9)

\[ f_1 = \frac{\sum_{i=1}^{n} y_i}{n} \] \hspace{1cm} (10)

where \( y \) represents the assignment state of all UAVs. If \( a_i \neq 0, y_i = 1, i \in \{1,2,\ldots,n\} \).

On the contrary, as long as there exists a UAV group that can not meet the tasks requirements, the constraint violation is calculated by the following equations.

\[ \text{Fr} = \begin{bmatrix} F_{r1}^1 & F_{r2}^1 & \cdots & F_{r1}^k \\ F_{r2}^1 & F_{r2}^2 & \cdots & F_{r2}^k \\ \vdots & \vdots & \ddots & \vdots \\ F_{rm}^1 & F_{rm}^2 & \cdots & F_{rm}^k \end{bmatrix} \] \hspace{1cm} (11)

\[ C = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} F_{rj}^i}{m \cdot k} \] \hspace{1cm} (12)

where \( \text{Fr} \) represents the requirement satisfaction of different tasks on different payloads. If the requirement of the task \( t_i \) on the \( j \)th payload is satisfied, \( F_{rj}^i = 0 \). Otherwise, \( F_{rj}^i = 1 \).

### 2.2. Path planning

In the task allocation process, for a feasible allocation scheme, the second objective function, i.e., the makespan \( f_2 \), should also be calculated.

Suppose that the payloads of UAV are equipped in the pod which can rotate 360°, and each group of UAVs flies to their assigned tasks in a leader-following mode. Considering that the trajectory of the UAV is constrained by curvature, the path planning problem of each group of UAVs can be regarded as a Dubins traveling salesman problem with neighborhood (DTSPPN) [6].

Each group of UAVs takes off from the runway of the base, detects all the tasks assigned to them, and then returns to the base and lands through the runway. In this situation, the departure and return positions of a UAV group are the same, but with opposite headings. Then, the path planning problem for each UAV group can be modeled as follows:

\[ \min D_i(S, P, H) = \sum_{j=0}^{m-1} d([s_j, p_j, h_j], [s_{j+1}, p_{j+1}, h_{j+1}]) \] \hspace{1cm} (13)

\[ + d([s_{m-1}, p_{m-1}, h_{m-1}], [s_0, p_0, \pi - h_0]) \]

where \( D_i \) (\( i \in \{1,2,\ldots,g\} \)) represents the length of the Dubins tour of the \( i \)th UAVs group, and \( g \) is the number of UAVs groups. \( S = [s_1, s_2, \ldots, s_{m-1}] \), \( P = [p_1, p_2, \ldots, p_{m-1}] \), \( H = [h_1, h_2, \ldots, h_{m-1}] \) represent the sequence of UAVs fly to tasks, and the visiting positions and headings at different tasks, respectively. \( s_0 \) represents the index of the base, and \( p_0 \) and \( h_0 \) are the position of the base and the direction of the runway. \( m \) is number of tasks assigned to the \( i \)th UAV group. \( d([\cdot, [\cdot]) \) is the Dubins distance function.

After the path planning of all UAV groups is completed, the makespan \( f_2 \) can be calculated by the following equation.

\[ f_2 = \frac{\max_{i=1}^{g} D_i}{D_{\text{max}}} , i=1,2,\ldots,g \] \hspace{1cm} (14)

where \( D_{\text{max}} \) is the longest Dubins tour, and it is used for the normalization of the makespan. It is calculated in the situation that all UAVs form a group to detect all tasks. Generally, the less the number of UAV group is, the larger the makespan is.

### 3. Problem solving

Based on the above, it can be seen that the heterogeneous UAV task planning problem can be regarded as a bi-level optimization problem. The first level optimizes the task allocation scheme, the second level optimizes the path planning scheme, and the inner and outer levels interact with each other.

#### 3.1. Algorithm for task allocation

In the first level, three “state-of-the-art algorithms” are considered to solve the task allocation problem of UAVs with heterogeneous payloads, and their characteristics are described as follows.
3.1.1. Overview of NSGA-II, DMOEA-εC and MOEA/D-AWA

(1) NSGA-II [9]: NSGA-II is the most popular multi-objective evolutionary algorithm. The dominance concept is used in NSGA-II to rank the solutions. Besides, it uses crowding distance to estimate the density of solutions near each solution. NSGA-II selects the best individual in both parent population and offspring population according to the crowding degree, so that the solution has a better distribution. NSGA-II uses the fast sorting method to construct the non-dominated solution set to reduce the complexity of the algorithm.

(2) DMOEA-εC [10]: DMOEA-εC is a new multi-objective evolutionary algorithm based on decomposition, which combines the ε-constraint method and decomposition strategy to re-model MOP under ε-constraint framework. The algorithm chooses an objective as the main objective, and assigns an upper bound vector to each subproblem, so that the MOP is decomposed into a series of single objective constrained optimization subproblems, and then uses the neighborhood information to optimize all subproblems, synergistically. In each generation, each solution corresponds to a subproblem, and the neighbor relationship between subproblems is defined according to the Euclidean distance between upper bound vectors. In addition, the main objective switching strategy, the matching mechanism from solution to subproblem and the matching mechanism from subproblem to solution are used to balance the convergence and diversity of the population in search process.

(3) MOEA/D-AWA [11]: The configuration of weight vectors is the key to the success of MOEA/Ds. Using uniformly distributed weight vectors usually leads to the deteriorated performance of MOEA/Ds on MOPs with irregular PFs. Thus, MOEA/D-AWA is proposed to deal with this issue by adaptively adjusting the weight vectors. It periodically deletes weight vectors by searching the overcrowded area, and adds new ones to search the sparse area of the current population with the help of an external population.

In order to solve the proposed problem with the above algorithms, the encoding, decoding, crossover and mutation operators are designed according to the characteristics of the problem.

3.1.2. Encoding and decoding

The decision variables of the task allocation problem are shown in Eqs. (6)-(7), and the encoding of this problem can be represented as follow:

\[ x = [G, A] \]

\[ = [r_1, r_2, \ldots, r_m, s_1, s_2, \ldots, s_{m-1}, a_1, a_2, \ldots, a_m] \tag{15} \]

The encoding consists of three parts: task sequence, task separation and UAV assignment, and the related description can be found in Subsection 2.1. For convenience of understanding, the following example illustrates how to decode \( x_i \) into a scheme.

\[ x_i = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

Figure 2 shows the whole process of decoding. In \( x_i \), there are 5 tasks, and the task sequence is \([1,3,5,2,4]\). Then, through the task separation vector \([0,1,0,1]\), it can be divided into three groups: \([1,3]\), \([5,2]\) and \([4]\). In addition, the assignment of 10 UAVs is \([0,2,5,1,3,2,4,0,1,5]\), which represents the assignments of \( u_1 \) to \( u_{10} \). Except for \( u_1 \) and \( u_5 \), other UAVs are assigned different tasks. By associating UAVs with the tasks in different task groups, the UAV groups are formed.

Based on the above, a complete scheme is formed. If this scheme is feasible, the objective function can be calculated by using Eqs. (10) and (14). Otherwise, the constraint violation is calculated by Eq.(12). It is noted that how to calculate the makespan (f2) by using Eq. (14) will be explained in Subsection 3.2.

3.1.3. Operators design

For NSGA-II, DMOEA-εC and MOEA/D-AWA algorithms, although their mechanisms are different, they all use crossover and mutation operators to generate new individuals. In this paper, the following crossover and mutation operators are designed according to the characteristics of the problem.

\[ x_i = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]
\[ x_j = [5,1,4,3,2,1,0,1,0,1,3,4,2,0,5,2,0,0] \]
\[ x_{\text{son}} = [1,3,5,4,2,0,1,1,0,0,1,5,4,2,2,5,0,0,5] \]

Figure 3. Diagram of the crossover operator

(1) Crossover operator

In the crossover operator, each dimension of two parents \( x_i \) and \( x_j \) crosses according to the crossover probability \( P_c \), as Fig. 3 shows. It is noted that in the task sequence part of the encoding, in order to ensure the integrality of the sequence, the elimination processing is carried out during the crossover.

(2) Mutation operators

A set of mutation operators are designed. The following explains the mutation process, and the yellow highlight indicates the tasks or UAVs being handled.
• Inter-group exchange of tasks (Fig. 4): randomly select two tasks in different groups and exchange them.

![Diagram of the inter-group exchange of tasks](image)

\[ x_{\text{son}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

\[ x_{\text{mut}} = [1,5,3,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

**Fig. 4.** Diagram of the inter-group exchange of tasks

• Inter-group movement of tasks (Fig. 5): randomly select a task from a certain group and move it randomly to another group.

![Diagram of the inter-group movement of tasks](image)

\[ x_{\text{son}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

\[ x_{\text{mut}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

**Fig. 5.** Diagram of the inter-group movement of tasks

• Merging of task groups (Fig. 6): randomly select two task groups and combine them.

![Diagram of the merging of task groups](image)

\[ x_{\text{son}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

\[ x_{\text{mut}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

**Fig. 6.** Diagram of the merging of task groups

• Decomposition of task group (Fig. 7): randomly select a task group, and randomly select one of the tasks from the group. It is important to note that the first task in the group cannot be chosen. Then, taking this task as the first task, a new task group is set up.

![Diagram of the decomposition of task group](image)

\[ x_{\text{son}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

\[ x_{\text{mut}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

**Fig. 7.** Diagram of the decomposition of task group

• Assignment change of UAVs (Fig. 8): randomly select a UAV that has been assigned a task, and change its task.

• Assignment cancellation of UAVs (Fig. 9): randomly select a UAV that has been assigned a task, and cancel its task.

As shown in Fig. 6, the mutation associated with task groups has a greater impact on the encoding, and the re-configuration of the part of encoding is needed. It should be noted that the independence of the assignment cancellation and increase of UAVs from the assignment change of UAVs is conducive to finding a more comprehensive PF when the number of UAVs is large, which makes algorithms more likely to generate a scheme with a change in the number of UAVs.

3.2. Algorithm for path planning

In order to calculate the makespan \( f_2 \), first, it is needed to calculate the task completion time of each UAVs group, respectively. As described in Subsection 2.2, the path planning for each UAV group is regarded as a DTSPN. Then, inspired by the study [12], a sampling-based heuristic method is adopted to solve the problem. The samplings of positions and headings of a certain UAVs group on each task are shown in Fig. 10.

![Diagram of the assignment increase of UAVs](image)

\[ x_{\text{son}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

\[ x_{\text{mut}} = [1,3,5,2,4,0,1,0,1,0,2,5,1,3,2,4,0,1,5] \]

**Fig. 10.** Diagram of the assignment increase of UAVs

As shown in these figures, the mutation associated with task groups has a greater impact on the encoding, and the re-configuration of the part of encoding is needed. It should be noted that the independence of the assignment cancellation and increase of UAVs from the assignment change of UAVs is conducive to finding a more comprehensive PF when the number of UAVs is large, which makes algorithms more likely to generate a scheme with a change in the number of UAVs.
Based on the above, the makespan ($f_2$) can be obtained by using Eq. (14).

4. Computational experiments

Random simulation instances with two scales are designed, including $n = 10, m = 10$ and $n = 20, m = 20$. $n$ is the number of UAVs, and $m$ is the number of tasks. Each UAV is equipped with no more than four sensors at random, and the detection success rates of these payloads on different tasks are set as 0.8 + 0.2 * rand. \texttt{rand} is a random real number between 0 and 1. The task requirements for each payload are set as 0.9.

The parameter setting of NSGA-II [9], DMOEA-$\varepsilon C$ [10] and MOEA/D-AWA [11] are referred to the literatures, except for the population size ($NP = 50$), the probability of the crossover and mutation ($P_c = 0.5, P_m = 0.2$), and the reference points $(0, 0; 1, 1)$. All the algorithms perform the same number of objective function evaluations ($NFE = 50000$), and run 20 times. Then, results are analyzed by using the inverted generational distance ($IGD$) and the hypervolume ($HV$) [14, 15], as Tab. 1 shows. Figure 14 shows the PFs with the best $IGD$ in-

dex of the three algorithms on different instances.

As can be seen from Tab. 1 and Fig. 14, the DMOEA-$\varepsilon C$ algorithm shows obvious advantages in solving the task planning problem of UAVs with heterogeneous payloads, and it can provide high quality and diversified planning schemes for commanders.

5. Conclusion

This paper studies the multi-objective task planning of UAVs with heterogeneous payloads, and a bi-level solution scheme is proposed. Then, the comparison of the performance of NSGA-II, DMOEA-$\varepsilon C$, and MOEA/D-AWA algorithms in this problem are presented. These algorithms are tested with multiple random instances with different scales. Results indicate that DMOEA-$\varepsilon C$ outperforms NSGA-II and MOEA/D-AWA algorithms in most of the instances and can provide more reliable planning scheme for commanders.

References:


Table 1. Statistical results about different algorithms in solving randomly generated instances

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10-10-1</td>
<td>(0.0573, 0.0956, 0.02/4, 0.01/67)</td>
<td>(0.0426, 0.0818, 0.02/02, 0.01/93)</td>
<td>(0.0626, 0.1212, 0.02/61, 0.02/57)</td>
</tr>
<tr>
<td>10-10-2</td>
<td>(0.02/41, 0.06/20, 0.00/01, 0.01/54)</td>
<td>(0.02/15, 0.05/40, 0.00/09, 0.01/19)</td>
<td>(0.04/46, 0.18/07, 0.00/12, 0.04/29)</td>
</tr>
<tr>
<td>10-10-3</td>
<td>(0.0676, 0.08/20, 0.01/83, 0.02/16)</td>
<td>(0.06/44, 0.08/28, 0.01/67, 0.01/74)</td>
<td>(0.07/89, 0.13/16, 0.05/29, 0.01/62)</td>
</tr>
<tr>
<td>20-20-1</td>
<td>(0.0632, 0.08/08, 0.04/76, 0.00/95)</td>
<td>(0.0632, 0.08/60, 0.04/71, 0.01/13)</td>
<td>(0.07/23, 0.09/96, 0.03/93, 0.01/72)</td>
</tr>
<tr>
<td>20-20-2</td>
<td>(0.0646, 0.07/05, 0.02/58, 0.01/08)</td>
<td>(0.04/81, 0.06/98, 0.01/90, 0.01/40)</td>
<td>(0.05/07, 0.06/43, 0.03/72, 0.00/73)</td>
</tr>
<tr>
<td>20-20-3</td>
<td>(0.05/90, 0.08/43, 0.07/16, 0.01/25)</td>
<td>(0.04/92, 0.06/70, 0.03/39, 0.00/09)</td>
<td>(0.07/36, 0.10/12, 0.03/64, 0.01/61)</td>
</tr>
</tbody>
</table>

| (a) 10-10-1   | | | |
| (b) 10-10-2   | | | |
| (c) 10-10-3   | | | |
| (d) 20-20-1   | | | |
| (e) 20-20-2   | | | |
| (f) 20-20-3   | | | |

Fig. 14. PFs of three algorithms on different instances