2018 IEEE International Conference on Systems, Man, and Cybernetics

An Estimation of Distribution Algorithm for Multi-robot Multi-point Dynamic Aggregation Problem

Bin Xin*, Shiqing Liu, Zhihong Peng, Guanqiang Gao
School of Automation
State Key Laboratory of Intelligent Control and Decision of Complex Systems
Beijing Advanced Innovation Center for Intelligent Robots and Systems
Beijing Institute of Technology
Beijing 100081, China
Email: *brucebin@bit.edu.cn; liushiqingbit@163.com; peng@bit.edu.cn; leon_gao@outlook.com

***Abstract*—Multi-Point Dynamic Aggregation (MPDA) is a novel task model for describing the process of multiple robots performing time-variant tasks. In the MPDA problem, several task points are located in different places and their states change over time. Multiple robots aggregate to these task points and execute the tasks cooperatively to make the states of all the task points change to zero. The task planning of MPDA is a typical NP-hard combinatorial optimization problem. Estimation of Distribution Algorithms (EDA) are evolutionary techniques based on probabilistic models. In this paper, a permutation-based EDA is proposed to solve the task planning problems in MPDA. The algorithm uses K-means clustering to update its probabilistic model which follows the multi-modal Gaussian distribution. Experimental results show that the proposed algorithm outperforms other compared methods in solving the task planning problems of MPDA.**

I. Introduction

The multi-robot system is widely applied to many real-life problems, such as search and exploration, disaster rescue and forest fire fighting. These complex tasks have some characteristics in common. Usually, a single robot is not competent to fulfill the task, and multiple robots are required for collaborative execution. The state of the task points varies with time, while the variability of state is affected by the task planning scheme of the multi-robot system. Take forest fire fighting as an example. Ignition points are scattered in the environment, with different fire intensities and increasing rates because of the wind. Each fire fighting robot has its own initial position and access order of ignition points. The robots assemble to their respective mission points and extinguish the fire cooperatively. After the fire is suppressed, the robot will move to the next ignition point and repeat the execution, until all the fires are extinguished.

A novel combinatorial optimization problem, called Multi-Point Dynamic Aggregation (MPDA), arises from the cooperation of multi-robots in complex tasks distributed in an extensive area [1]–[4]. In the MPDA task, a number of task points are located in different places and their states change over time. Multiple robots have their own initial position and execution abilities. Robots will visit their ordered task points and execute the task cooperatively. The tasks are completed when the state of each task point reaches a desired value. The goal of MPDA is to complete all the tasks in an earliest time.

MPDA has resemblance to TSP and VRP. They are all permutation-based combinatorial optimization models abstracted from practical problems. However, there are two main differences between MPDA and others. First, in TSP and VRP, there is only one executor responsible for each task, while in MPDA, each task may be cooperatively performed by multiple robots simultaneously. Secondly, the state of each task point in MPDA is time-varying. The task completion time depends not only on the time when robots reach the task point, but also on the change of the state of each task point. Meanwhile, the change of the state is affected by the task planning scheme. This coupling relationship does not exist in TSP or VRP.

The MPDA problem has attracted some attention from researchers. In [5], MPDA was applied to formulate the multi-robot motion planning problem in the cooperative multi-area coverage. A one-dimensional variant of the MPDA model is the multi-robot persistent monitoring problem. An optimal control framework was proposed by using infinitesimal perturbation analysis and a gradient-based algorithm to solve the multi-robot persistent monitoring problem [6]. A distributed path planning algorithm with the strategy of receding horizon was proposed to solve the motion planning problem of multiple robots in MPDA [1]. A meta-heuristic method with decoupling strategy is presented to solve the path planning problem of messenger UAVs in MPDA task [7].

The dynamic characteristics of task points result in a substantial increase in the complexity of the problem and the solving difficulty. Therefore, exact approaches are typically only used for smaller problem instances, and meta-heuristic algorithms such as Estimation of Distribution Algorithm (EDA)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673058, in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1609214, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621063.

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00140

are more suitable for solving the MPDA task planning problem. EDA is a kind of evolutionary algorithms which has been developed to solve permutation-based problems because it can take advantage of the distribution characteristics of potential solutions to guide the evolution direction [8]–[15] .

This paper proposes a new EDA based on relative order model to solve permutation-based optimization problems in the task planning of MPDA. The contributions of this paper are summarized as follows:

- Forest fire fighting is abstracted and modeled as a multi-permutation combinatorial optimization problem called MPDA.
- An effective coding/decoding strategy is proposed for MPDA.
- An EDA is proposed for solving the task planning problem of MPDA. Experimental results show that the proposed EDA can find high-quality solutions and outperforms specially-designed Genetic Algorithm (GA) in most cases.

The rest of the paper is organized as follows. Section II defines the MPDA task. In Section III, the proposed EDA is described in detail. The experimental settings, results and discussion are reported in Section IV. Section V concludes the paper.

II. Problem Formulation

In MPDA, a number of task points are located at different places and their states change over time. The physical state S (e.g., the fire intensity in a forest fire fighting case) of each point considered in this paper is assumed to follow an exponential law. It can be described by the equation (1):

$$S = S_0 e^{\alpha t} \tag{1}$$

S in equation (1) represents the physical state of a task point, S_0 is the initial state value, t is time (its initial value is set to 0), α is a growth factor and $\alpha > 0$, which means that the physical state of the task point is rising over the time at an exponential rate. It is provided that the task is completed when the state S of the task is changed to zero (allowing some errors). Consider the MPDA task in a two-dimensional plane where multiple robots aggregate from their respective start points to their task points. Each robot begins to perform its task when it arrives at the task point. In the paper each robot will give the task point a negative growth coefficient $-\beta$ (coefficients for all robots are the same) so that after the robot reaches its corresponding task point, the state of the task point becomes:

$$S = S_1 e^{(\alpha-\beta)t} \tag{2}$$

In equation (2), S_1 represents the state value of the task point when the robot reaches the task point at time t_1. It is known from the equation (2) that the state growth of the task point will start to slow down from time t_1. In addition, a number of robots can aggregate to one task point together to accomplish the goal of the task which is to drive S to zero. The cooperation effect is simplified to the superposition of a plurality of $-\beta_i$ so that if there are two robots aggregating to the task point , the state S will change by equation (3) after both of the robots reaching the task point.

$$S = S_1 e^{(\alpha-\beta_1-\beta_2)t} \tag{3}$$

Equation (3) shows that multiple robots can accomplish the task faster. The state model described above indicates that the state of the task point is changing and the environment of MPDA may also change. So the behavioral characteristics of the robots need to follow the dynamic changes so as to achieve the goal of the task. And this requires that both task planning and decision making of the robots be accomplished so that each robot can coordinate its movement with each other in order to complete the task faster and better. MPDA can be used to describe a wide range of problems, such as forest fire fighting, disaster relief, target monitoring and so on.

In the MPDA task planning problem studied in this paper, it is assumed that the multi-robot system consists of m intelligent robots. All of the robots can move freely and execute the task in the two-dimensional work space. Each robot has its own initial position, velocity and executive capacity. The robot with a larger executive capacity β has a greater impact on the state of the task point.

The model of multiple robots and task points in this article is based on the following premises and assumptions:

1) There are m robots $(rob_i, i = 1, 2, ..., m)$ which can be treated as particles.
2) There are n task points $(task_j, j = 1, 2, ..., n)$ with different locations.
3) Each robot has its own initial position P_0, velocity v and executive capacity β.
4) Each robot knows its own start and end points but does not know those of other robots.
5) There are no obstacles in the work space, and collision avoidance is beyond consideration. Each robot can move freely with a fixed velocity.
6) The robot will no longer go to the task point where the task has been completed.
7) The robot can only move to other task points after completing the current task.
8) A complete scheme of the task planning is planned ahead of the execution.

In a task planning problem of MPDA with $m = 4$ robots and $n = 4$ task points, the problem can be formulated as follows:

$$T = \begin{bmatrix} x_1 & y_1 & \alpha_1 & S_{01} \\ x_2 & y_2 & \alpha_2 & S_{02} \\ x_3 & y_3 & \alpha_3 & S_{03} \\ x_4 & y_4 & \alpha_4 & S_{04} \end{bmatrix}, R = \begin{bmatrix} x_{01} & y_{01} & v_1 & \beta_1 \\ x_{02} & y_{02} & v_2 & \beta_2 \\ x_{03} & y_{03} & v_3 & \beta_3 \\ x_{04} & y_{04} & v_4 & \beta_4 \end{bmatrix} \tag{4}$$

In equation (4), Matrix T represents the information of task points, in which (x_j, y_j) are the horizontal and vertical

coordinates of $task_j$, α_j is the growth factor, and S_{0j} is the initial state value of the task. Matrix R represents the information of the robots, in which (x_{0i}, y_{0i}) are the initial horizontal and vertical coordinates of rob_i, v_i is the velocity, and β_i is the executive capacity of the robot.

The aim is to minimize the following objective function:

$$F(X) = \max_j \{t_j(X) \mid T, R\} \tag{5}$$

where t_j is the time instant when $task_j$ is completed, T and R denote the parameters of tasks and robots respectively, and $\max_j \{t_j(X) \mid T, R\}$ implies the moment when all the tasks are completed.

III. An Estimation of Distribution Algorithm for MPDA Task

A. Coding / Decoding Strategy

In an MPDA task planning problem with m robots and n task points, a feasible solution is encoded as an $m \times n$ matrix X:

$$X = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{bmatrix} \tag{6}$$

where the ith row $[p_{i,1}\ p_{i,2}\ \cdots\ p_{i,n}](i = 1, 2, ..., m)$ is an integer permutation of n task points, which stands for the visiting route of rob_i. For example, permutation $[2\,4\,3\,1]$ means that the robot will first visit $task_2$ and execute the task, then move to $task_4$, $task_3$ and finally move to $task_1$ if necessary.

In the decoding strategy, if the state of one task has dropped to zero before the robot arrives, the task will be deleted from the visiting route of this robot. Robots cannot leave their current task point until the task is finished. In a feasible solution, different robots may have the same visiting route, but each task can appear only once in a permutation. The decoding strategy is described as **Algorithm 1**.

In the algorithm, the arriving time matrix T_{arr} and the completion time matrix T_{com} is described as:

$$T_{arr} = \begin{bmatrix} t^a_{1,1} & t^a_{1,2} & \cdots & t^a_{1,n} \\ t^a_{2,1} & t^a_{2,2} & \cdots & t^a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ t^a_{m,1} & t^a_{m,2} & \cdots & t^a_{m,n} \end{bmatrix} \tag{7}$$

$$T_{com} = \begin{bmatrix} t^c_{1,1} & t^c_{1,2} & \cdots & t^c_{1,n} \\ t^c_{2,1} & t^c_{2,2} & \cdots & t^c_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ t^c_{m,1} & t^c_{m,2} & \cdots & t^c_{m,n} \end{bmatrix} \tag{8}$$

where $t^a_{i,j}(i = 1, 2, ..., m; j = 1, 2, ..., n)$ represents the time instant when rob_i arrives at $task_j$ in the order of the matrix X, and $t^c_{i,j}(i = 1, 2, ..., m; j = 1, 2, ..., n)$ represents the time instant when $task_j$ is finished by rob_i. $t^a_{i,j}$ is set to -1 when the task has been finished. An event matrix T_h describes the execution process of all the tasks:

$$T_h = \begin{bmatrix} robNum_1 & taskNum_1 & arrTime_1 & comTime_1 \\ robNum_2 & taskNum_2 & arrTime_2 & comTime_2 \\ \vdots & \vdots & \vdots & \vdots \\ robNum_\delta & taskNum_\delta & arrTime_\delta & comTime_\delta \end{bmatrix} \tag{9}$$

where $robNum_k$ and $taskNum_k(k = 1, 2, ..., \delta)$ represent the number of the robot and the task in the kth event separately, while $arrTime_k$ and $comTime_k$ represent the arriving moment and completion moment separately. T_h consists of δ events, and each event describes the arriving of a robot or the completion of a task.

Algorithm 1 Decoding Strategy

```
Input: R, T, X
Output: F(X)
Calculate the arriving time matrix T_arr ;
Calculate the completion time matrix T_com;
Initialize the number of the event δ = 1;
for each(i, j) with(t^a_{i,j} ≠ −1) do
  (i*, j*) ← argmin(t^a_{i,j})
  if (i, j) = (i*, j*) then
    t^a_{i,j} = −1;
  end if
  Add the current event into matrix T_h;
  if δ ⩾ 2 then
    for k = 1 to δ − 1 do
      if k = δ then
        if arrTime_δ < comTime_k then
          Update T_arr and T_com;
        else
          δ = δ − 1;
          Update T_h, T_arr and T_com;
        end if
      end if
    end for
  end if
  δ = δ + 1;
end for
```

B. EDA for MPDA

Estimation of distribution algorithm is a class of evolutionary computing paradigms [8], [11], [16]. In each generation, EDA estimates the overall distribution of the parent solutions, and a probabilistic model is updated by using the information of the distribution of current solutions. Offspring is generated by sampling the constructed model. The evolution process continues until a stopping criterion is met (e.g., the current best objective function value is smaller than a given value or the number of generations is equal to a given maximum value). Modeling paradigms include statistical methods [17], probability mechanisms [18], and machine learning approaches [19]–

[21]. In this paper, three main operators of EDA for MPDA are described as follows:

Selection. In order to retain the characteristics of potential solutions, truncation selection is used to select N_{adv} solutions from parent population ($N_{adv} < N_p$). In the strategy, individuals are sorted in an ascending order according to their objective function values, and then the top N_{adv} individuals are selected to form a new population.

Modeling. For each robot and for each task point, we first count the sequence number that the task point appears in the robot's task planning scheme. The statistics imply a probability that the task point appears in each sequence number. Then K-means clustering is used to divide the statistical data of each task into k categories (k is set to two in this paper for simplification). For each category, we calculate the mean value and the standard deviation of the statistical data, and fit them into a Gaussian distribution. Therefore, the probabilistic modal of each robot consists of k Gaussian distributions. After the process, we can get a probabilistic model in the form of multi-modal Gaussian distribution for each robot.

Sampling. New solutions are generated by sampling the constructed probabilistic model. In this paper, a sampling method called relative order sampling is used to generate new feasible individuals. Here is an example about how to generate a permutation of n tasks for a robot. First, for each task, we sample its multi-modal Gaussian distribution model and get a decimal number, which stands for the relative visiting order of the task in the visiting route of the robot. Then, after getting all of the n decimals, we sort them in an ascending order to get the absolute order of the tasks, which is stored as a row vector in a new solution matrix. For example, there are four tasks and three robots in MPDA, and the sampling and decoding matrices are

$$X_{sample} = \begin{bmatrix} 2.8 & -0.2 & 4.3 & 3.5 \\ 3.1 & 4.3 & 2.0 & 1.1 \\ -0.3 & 1.5 & 3.9 & 2.8 \end{bmatrix}$$

$$X_{decode} = \begin{bmatrix} 2 & 1 & 4 & 3 \\ 4 & 3 & 1 & 2 \\ 1 & 2 & 4 & 3 \end{bmatrix}$$

where [2.8 -0.2 4.3 3.5] in X_{sample} changes into [2 1 4 3] in X_{decode} by sorting-based decoding. It means that the robot will first move to $task_2$, and then move to $task_1$, $task_4$ and $task_3$.

We propose an EDA based on the above operators for solving MPDA problem. The process is given as follows:

Step 1: Randomly generate N_p feasible solutions to form an initial population.

Step 2: Evaluate the individuals in the current population by the decoding strategy in **Algorithm 1**.

Step 3: Select N_{adv} solutions from parent population by using the truncation selection ($N_{adv} < N_p$). The selected individuals constitute a dominant population.

Step 4: Update the probabilistic model of the dominant population based on K-means clustering and multi-modal Gaussian distribution.

Step 5: Sample the constructed probabilistic model to get new solutions and form a new population.

Step 6: If the stopping criterion is satisfied, terminate the calculation. Otherwise, turn to step 2.

C. Computational Complexity Analysis

The proposed EDA is used to solve an MPDA task planning problem with m robots and n tasks. In each iteration, there are N_p individuals in the population. Table I shows the computational complexity of the processes in EDA.

TABLE I
COMPUTATIONAL COMPLEXITY OF THE PROCESSES IN EDA

No.	Process	Computational Complexity
1	Initialization	$O\left(N_p * m * n^2\right)$
2	Evaluation	$O\left(m^2 n^2\right)$
3	Selection	$O\left(N_p \log(N_p)\right)$
4	Modeling	$O\left(N_p + \log(N_p)\right)$
5	Sampling	$O\left(\log(N_p * m * n) + N_p * m * n \log(n)\right)$

The worst-case time complexity of the algorithm can be approximately expressed by:

$$O(mn(mn + N_p n + N_p^2 \log(n))) \tag{10}$$

IV. EXPERIMENTS AND RESULTS

This section is devoted to the performance investigation of the proposed algorithm. First of all, an MPDA test-case generator is presented to produce instances of different scales. Then, we briefly describe the algorithms for comparison and the parameter settings. The final part illustrates the experimental results. All experiments were carried out in MATLAB_R2016b environment on a PC with Intel Xeon E5 CPU 2.60GHz and 32GB RAM.

A. Test-Case Generator for MPDA

Since there are no benchmark problems for MPDA, we first developed a test case generator in order to test the performance of different algorithms in solving MPDA problems of varied scales. Matrix T and R in equation (4) formulate an MPDA task with m robots and n tasks. Given m and n, the generator will provide the matrix T and R, which include all the essential parameters for an MPDA problem. In particular, the objective function value is defined as infinity in deadlock cases.

1) The generation of T:

The work space of a two-dimensional MPDA problem is defined as a rectangle with a side-length l. In matrix T, x_j and y_j describe the coordinates of $task_j (j = 1, 2, \cdots, n)$, which are randomly generated from the work space. Each of the growth factor α_j is randomly generated from $(0.1, 0.9)$. The initial value of the physical state S_{0j} is generated as uniformly distributed random integers in the range from 5 to 20.

2) The generation of R:

The initial coordinates of robots, x_{0i} and y_{0i} , are generated by the same method as x_j and y_j. The velocity v_i is set to 1 uniformity. β_i is randomly generated from $(0.1, 0.9)$.

B. Algorithms for Comparison

1) Genetic Algorithm (GA):

GA is a representative evolutionary algorithm for solving combinatorial optimization problem. GA mainly consists of three operators: selection, crossover and mutation. In each iteration of GA, we use tournament selection operator to select dominant individuals. Two crossover operators are selected with the same probability to generate new individuals. One is crossover by lines and the other is reconstruction after crossover by columns. Then two-point exchange mutation are performed. The probabilities of crossover and mutation are P_x and P_m respectively. The algorithm terminates when the stopping criteria is satisfied.

2) Random Sampling (RS) method:

In RS, each sampling is achieved by a pure random permutation of the integers from 1 to n, representing the visit order of a robot. m permutations are sampled to generate a solution X.

C. Parameter Settings

A total of 15 random instances with different scales are generated to test the universality and scalability of the algorithm. There are three small-scale cases with $m, n \in (1, 5)$, eight cases with $m, n \in (5, 20)$ and four complex cases with $m, n \in (20, 30)$. To ensure the effectiveness of solutions, it is assumed that $m \geqslant n$. T and R for each case are generated by the method of section A. The stopping criteria is defined as a maximum number of function evaluations (Max_FES).

GA and EDA require two parameters in common (i.e. N_p and Max_FES). We performed multiple runs to determine the preferable parameter values. Based on the running results, the parameter settings are empirically determined and shown in Table II .

TABLE II
PARAMETER SETTINGS

Parameter	Value
Population Size (N_p)	100
Subpopulation Size (N_{adv})	$0.3 * N_p$
Crossover Probability (P_x)	0.7
Mutation Probability (P_m)	0.1
Maximum Function Evaluations (Max_FES)	10000

In consideration of the stochastic characteristic of the algorithms, each method ran 20 times independently for each instance, and the results are statistically analyzed. For fair comparison, the same stopping criteria is adopted in GA and EDA, while RS generates Max_FES random samplings in each case.

D. Results and Discussion

This section gives the experimental results of the three algorithms in solving MPDA problem with different scales. Table III shows the results of 20 independent runs of each algorithm in the form of mean values and standard deviations.

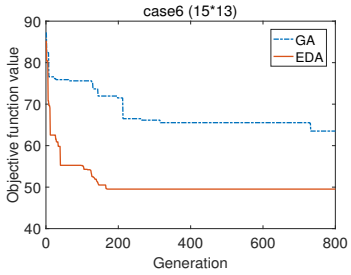

Fig. 1. Convergence Curves of GA and EDA in case6

TABLE III
STATISTICAL RESULTS OF 20 INDEPENDENT RUNS FOR DIFFERENT ALGORITHMS IN 15 CASES

No.	GA	EDA	RS
1	**80.931 ± 0.000**	**80.931 ± 0.000**	**80.931 ± 0.000**
2	**84.647 ± 0.000**	87.435 ± 1.013	87.494 ± 0.621
3	**34.611 ± 0.000**	**34.611 ± 0.000**	**34.611 ± 0.000**
4	41.423 ± 0.429	**40.732 ± 0.736**	47.393 ± 1.315
5	29.897 ± 0.387	**28.005 ± 0.653**	35.953 ± 1.017
6	61.926 ± 1.332	**53.552 ± 2.794**	77.179 ± 1.830
7	42.928 ± 1.025	**36.814 ± 1.283**	56.057 ± 0.942
8	49.763 ± 1.284	**41.191 ± 2.019**	60.968 ± 0.903
9	46.964 ± 1.398	**36.861 ± 2.070**	61.445 ± 1.713
10	64.375 ± 1.564	**55.552 ± 2.859**	74.390 ± 1.158
11	25.862 ± 0.435	**23.312 ± 0.687**	32.042 ± 0.730
12	60.188 ± 1.190	**56.223 ± 1.426**	72.690 ± 1.634
13	72.231 ± 1.283	**58.476 ± 1.781**	87.479 ± 2.432
14	71.917 ± 1.852	**56.461 ± 1.823**	91.819 ± 2.152
15	61.382 ± 1.943	**52.750 ± 1.927**	76.261 ± 1.515

1) The Performance of EDA for MPDA: It can be drawn from Table III that the proposed EDA outperforms others in general, especially with the increase of the number of task points. To be specific, the mean objective value obtained by EDA is on average 13.2% lower than that by GA, and 38.8% lower than that by RS.

In case 1 and 3, all of the three methods converge to the optimal solution. This is because the small size of the instance makes it easy to cover all feasible solutions. For medium-scale cases (case 4, 5 and 11), EDA and GA have similar performances. GA and RS may generate a better solution than EDA with a finite number of sampling. However, EDA has obviously higher quality in contrast to GA and RS in complex cases. The quality of solutions generated by RS and GA is unstable and decrease with the growth of the problem size. This is because EDA can update the probabilistic model according to the distribution of dominant solutions, while the iterative direction of GA is randomly generated.

In terms of scalability, EDA-based algorithm can solve complex cases of MPDA problems within a fixed computation cost, while it is almost impossible for GA and RS to find an acceptable solution effectively.

2) Convergence Analysis: Figure 1 provides the convergence curves of GA and EDA in case 6. It can be drawn

from the figure that EDA has a faster convergence speed than GA.

3) Running Time Analysis: Table IV shows the running time of the three algorithms. For each case, the mean value and standard deviation of twenty independent running times are calculated. Among the three methods, RS takes the shortest time, followed by GA, while EDA takes the longest time. It can be concluded from the results that EDA has the greatest computational complexity among the comparison algorithms. It is mainly attributed to the modeling and sampling process, consisting of the clustering and sorting operators.

TABLE IV
RUNTIME PERFORMANCE OF DIFFERENT ALGORITHMS (SEC.)

No.	GA	EDA	RS
1	1.4857 ± 0.0462	2.9518 ± 0.2157	0.7803 ± 0.0041
2	2.3976 ± 0.1598	5.2239 ± 0.3862	1.1797 ± 0.0296
3	2.4889 ± 0.3101	5.2393 ± 0.3114	1.1833 ± 0.0023
4	3.6693 ± 0.3429	8.1903 ± 0.6301	1.4617 ± 0.0390
5	6.7362 ± 0.6382	14.5236 ± 1.0640	3.0949 ± 0.0147
6	12.9583 ± 1.0264	26.9653 ± 1.3758	4.8953 ± 0.0305
7	17.4673 ± 1.6219	36.6295 ± 2.6613	7.8284 ± 0.0198
8	28.3974 ± 2.1324	57.1677 ± 3.8209	11.8061 ± 0.1428
9	32.4633 ± 2.9633	65.2624 ± 4.8654	15.3404 ± 0.6775
10	51.2390 ± 3.8147	98.1860 ± 6.4070	23.5229 ± 0.0427
11	69.5863 ± 6.4218	124.3808 ± 7.8429	32.0420 ± 0.0714
12	89.8471 ± 8.1657	155.0949 ± 12.8822	43.1199 ± 0.2136
13	114.7672 ± 9.5233	188.0666 ± 15.7042	55.7579 ± 0.2317
14	143.6200 ± 13.4076	225.4500 ± 19.0281	70.9565 ± 0.3577
15	176.2143 ± 16.0194	272.8830 ± 22.1947	88.5214 ± 0.6035

Overall, both GA and EDA can find feasible and high-quality solutions. The proposed EDA with K-means clustering and the multi-modal Gaussian distribution outperforms GA in most cases.

V. CONCLUSION

In this paper, forest fire fighting is abstracted and modeled as a multi-permutation combinatorial optimization problem called MPDA. A coding/decoding strategy was proposed for MPDA, where the feasible solution of MPDA is encoded in the form of a multi-permutation matrix. We also presented an EDA for solving the task planning problem of MPDA. In the proposed algorithm, a multi-modal Gaussian distribution is adopted for describing the probabilistic model, and K-means clustering is used to update the model. The performance of the algorithm is compared with GA and RS. Experimental results show that the proposed EDA outperforms GA and RS in general, and has a significant advantage with the increase in the number of task points.

Future research would be targeted on how to incorporate priori knowledge into the design of the probabilistic model of EDA.

REFERENCES

[1] B. Xin, Y. G. Zhu, Y. L. Ding, and G.-Q. Gao, "Coordinated motion planning of multiple robots in multi-point dynamic aggregation task," vol. 2016-July, Kathmandu, Nepal, 2016, pp. 933 – 938.

[2] Q. Yang, H. Fang, J. Chen, Z. P. Jiang, and M. Cao, "Distributed global output-feedback control for a class of euler lagrange systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4855–4861, Sept 2017.

[3] H. Fang, C. Shang, and J. Chen, "An optimization-based shared control framework with applications in multi-robot systems," *Science China Information Sciences*, vol. 61, no. 1, p. 014201, Dec 2017.

[4] J. Zeng, L. Dou, and B. Xin, "Multi-objective cooperative salvo attack against group target," *Journal of Systems Science and Complexity*, vol. 31, no. 1, pp. 244–261, 2018.

[5] B. Xin, G. Q. Gao, Y. L. Ding, Y. G. Zhu, and H. Fang, "Distributed multi-robot motion planning for cooperative multi-area coverage," in *2017 13th IEEE International Conference on Control Automation (ICCA)*, July 2017, pp. 361–366.

[6] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," in *52nd IEEE Conference on Decision and Control*, Dec 2013, pp. 6886–6891.

[7] Y. L. Ding, B. Xin, J. Chen, H. Fang, Y. G. Zhu, G. Q. Gao, and L. H. Dou, "Path planning of messenger uav in air-ground coordination," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8045–8051, 07 2017.

[8] L. Pedro and J. A. Lozano, Eds., *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science and Business Media, 2001.

[9] J. Ceberio, E. Irurozki, A. Mendiburu, and J. A. Lozano, "A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 103–117, Apr 2012.

[10] H. Liu, L. Gao, and Q. Pan, "A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4348 – 4360, 2011.

[11] J. A. Lozano, Ed., *Towards a new evolutionary computation: advances on estimation of distribution algorithms*. Springer Science and Business Media, 2006.

[12] J. Bassem, E. Mansour, and S. Patrick, "An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems," *Computers and Operations Research*, vol. 36, no. 9, pp. 2638 – 2646, 2009.

[13] M. Ayodele, J. McCall, and O. Regnier-Coudert, "Rk-eda: A novel random key based estimation of distribution algorithm," in *Parallel Problem Solving from Nature – PPSN XIV*, J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter, Eds. Cham: Springer International Publishing, 2016, pp. 849–858.

[14] M. Ayodele, J. McCall, O. Regnier-Coudert, and L. Bowie, "A random key based estimation of distribution algorithm for the permutation flowshop scheduling problem," in *Proceedings of 2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 2364–2371.

[15] J. Chen, X. Zhang, B. Xin, and H. Fang, "Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 959–972, April 2016.

[16] R. Santana, P. Larrañaga, and J. A. Lozano, "Research topics in discrete estimation of distribution algorithms based on factorizations," *Memetic Computing*, vol. 1, no. 1, pp. 35–54, Mar 2009.

[17] X. Zhong and W. Li, "A decision-tree-based multi-objective estimation of distribution algorithm," in *Proceedings of 2007 International Conference on Computational Intelligence and Security (CIS 2007)*, Dec 2007, pp. 114–11/8.

[18] P. A. Bosman and D. Thierens, "Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 259 – 289, 2002.

[19] H. Tang, V. A. Shim, K. C. Tan, and J. Y. Chia, "Restricted boltzmann machine based algorithm for multi-objective optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–8.

[20] L. Marti, J. Garcia, A. Berlanga, and J. M. Molina, "Solving complex high-dimensional problems with the multi-objective neural estimation of distribution algorithm," in *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, GECCO-2009*, Montreal, QC, Canada, 2009, pp. 619 – 626.

[21] Q. Zhang, A. Zhou, and Y. Jin, "Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, Feb 2008.