A Hybrid Differential Evolution and Estimation of Distribution Algorithm for the Multi-Point Dynamic Aggregation Problem*

Rong Hao, Jia Zhang, Bin Xin†
School of Automation, Beijing Institute of Technology; Beijing Advanced Innovation Center for Intelligent Robots and Systems
Beijing, China
brucebin@bit.edu.cn

Chen Chen, Lihua Dou
School of Automation, Beijing Institute of Technology; State Key Laboratory of Intelligent Control and Decision of Complex Systems
Beijing, China

ABSTRACT

The multi-point dynamic aggregation (MPDA) is a typical task planning problem. In order to solve the MPDA problem efficiently, a hybrid differential evolution (DE) and estimation of distribution algorithm (EDA) called DE-EDA is proposed in this paper, which combines the merits of DE and EDA. The DE-EDA has been applied to multiple MPDA instances of different scales, and compared with EDA and two versions of DE in convergence speed and solution quality separately. The results demonstrate the DE-EDA can solve the MPDA problem effectively.

CCS CONCEPTS

• **Applied computing** → **Operations research**; • **Computing methodologies** → Artificial inteligence;

KEYWORDS

Multi-point dynamic aggregation, Hybridization, Differential evolution, Estimation of distribution algorithm

ACM Reference Format:

Rong Hao, Jia Zhang, Bin Xin and Chen Chen, Lihua Dou. 2018. A Hybrid Differential Evolution and Estimation of Distribution Algorithm for the Multi-Point Dynamic Aggregation Problem. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205732

1 INTRODUCTION

Multi-point dynamic aggregation (MPDA) [1, 2] is a typical task model, which can be used to describe a wide range of complex task planning problems such as forest fire fighting, disaster relief, target surveillance and so on. Differential evolution (DE) [3], proposed by Kenneth Price and Rainer Storn, is very successful for global continuous optimization problem. Estimation of distribution algorithm (EDA) [4] is more effective to solve nonlinear optimization problems with coupled variables. As is well known, hybridization is effective to improve the performance of a single algorithm. In this research, a hybrid DE and EDA for MPDA, called DE-EDA, is proposed to improve the global searching ability and convergence speed of the algorithm.

2 MPDA PROBLEM

The MPDA problem contains multiple task points (every task point is represented in the below with TP) and unmanned vehicles (represented in the below with UV) randomly spread at different positions. The UVs are required to gather around these TPs to form a distribution according to the state information of each TP obtained by their sensors. When the UV reaches the neighborhood around the TP and begins to perform its task, the state of the TP will be changed. The target TPs of the UVs change according to the state information of the TPs, so as to make the state of each TP reach an expected value quickly. With the update of the state information, the UVs adaptively change the aggregation point until all tasks are completed.

In the MPDA problem, each UV can not perform the next task until the current task is completed. The solution of MPDA is expressed as the access orders of each UV to multiple TPs, and each UV is only allowed to access each TP at most one time. Assuming that the task planning scheme for the ith UV can be represented as x_i, the planning schemes for all UVs can be represented as $X = \{x_1, x_2, \cdots, x_m\}$. Therefore, solutions based on multiple permutations can be expressed as follows:

$$X = \{x_1, x_2, \cdots, x_m\} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{bmatrix}_{m\times n} \quad (1)$$

where x_i is a permutation of 1 to n, $s_{i,j} \in [1, 2, \cdots, n]$ denotes the number of the jth TP that the ith UV accesses, $i \in [1, 2, \cdots, m]$, $j \in [1, 2, \cdots, n]$. According to X and the attribute information of the UVs and the TPs, the arrival time and completion time of each UV to reach each target TP can be calculated.

The task execution scheme of multiple UVs can be obtained by solving the optimization problems in equation (2).

*Produces the permission block, and copyright information
†The corresponding author

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205732

$$\begin{aligned} &\arg\min_X \quad J(X) = \arg\min_X \quad \{\max_j\{t_j^*(X)|E\}\} \\ &s.t. \quad t_j^*(X) = \min\{t|x_j(t) \le \varepsilon\}, \forall j \in \{1,2,\cdots,n\} \end{aligned} \tag{2}$$

where $t_j^*(X)$ represents the completion time of task j, E denotes the environmental factor, and the earliest time when the state of the jth TP falls below the ε is the completion time of the task j.

3 HYBRID DE-EDA

In order to solve the MPDA problem efficiently, a hybrid DE and EDA, called DE-EDA, is proposed in this paper.

(1) *DE:* In mutation operation, a decisive factor R is used to determine which operator to choose to generate the mutant individuals $V_k(t)$, where $R \in [0,1]$. the detail is described as follows:

$$\begin{aligned} &if(rand < R) \quad V_k(t) = X_{r1}(t) + F*(X_{r2}(t) - X_{r3}(t)) \\ &else \quad V_k(t) = X_{best}(t) + F*(X_{r1}(t) + X_{r2}(t) - X_{r3}(t) - X_{r4}(t)) \end{aligned} \tag{3}$$

where $X_{r1}(t), X_{r2}(t), X_{r3}(t)$ and $X_{r4}(t)$ are randomly selected from the current population and $r1 \neq r2 \neq r3 \neq r4 \neq k$. $X_{best}(t)$ is the best solution of the current population, $F \in [0,1]$ is a scaling factor.

Each vector of the kth mutated individual is always a floating-point vector, which is infeasible. The random key method [5] is used to map the floating-point vector to a permutation vector.

(2) *EDA:* A univariate EDA named Univariate Marginal Distribution Algorithm (UMDA) is selected in this DE-EDA. For the MPDA, the model is constructed by using the probability matrix of the coding position for tasks performed by the UVs in each generation. For individual k, its probability model $P^k(t)$ is described as follows:

$$P^k(t) = \begin{bmatrix} P_{11}^k(t) & P_{12}^k(t) & \cdots & P_{1n}^k(t) \\ P_{21}^k(t) & P_{22}^k(t) & \cdots & P_{2n}^k(t) \\ \vdots & \vdots & \vdots & \vdots \\ P_{n1}^k(t) & P_{n2}^k(t) & \cdots & P_{nn}^k(t) \end{bmatrix}_{n\times n} \tag{4}$$

where $P_{ij}^k(t)$ denotes the probability that the ith TP is located at the jth position for individual k at generation t.

(3) *DE-EDA:* In the proposed algorithm, DE and EDA are combined in series, which means the offsprings are generated by DE and EDA alternately. And the reboot mechanism is used to prevent from being trapped in local optimum in the later evolution. The DE-EDA offspring generation scheme is described as Tabel 1.

Table 1: The DE-EDA Offspring Generation Scheme

Step 1: Select a population X_{sel} of size $Np*\alpha$, and build the probability model P by equation (4)
FOR i=1 : $(1-\alpha)*Np$ // α is the selection rate
Step 2: Sample from P and generate a new individual
Step 3: Evaluate the fitness value of the new individual
END FOR
Step 4: Get the offspring X by combining the X_{new} with X_{sel}
Step 5: Sort the fitness value of the population, and get the current optimal solution X_{best}
FOR i=1 : Np // Np denotes the Population size.
Step 7: Generate the mutation individual by equation (3)
Step 8: Deal with the mutation vectors using the random key
Step 9: Generate the trial individual through crossover
Step 10: Generate the offspring population $X(t+1)$ by selection operation
END FOR

4 EXPERIMENTS AND RESULTS

To measure the performance of our algorithm, the DE-EDA was applied to a plenty of instances of different scales, and was compared with two versions of DE and EDA in solution quality and time cost. The algorithm stops when the the number of objective function evaluations is equal to the maximum value $MaxFES$.

According to statistical results (see Tabel 2), we find that DE/rand/1/bin can find a better solution only in small-scale instances (like 5×5), and EDA can obtain a better solution for mid-scale instances (like 10×10), however, with high time cost. Therefore, our proposed DE-EDA has better performances in solution quality and convergence speed for most instances. These results indicated that the proposed hybrid algorithm DE-EDA can effectively solve the MPDA problem.

Table 2: Experimental Results

Scale	Algorithm	Mean±std	aver. runtime (sec.)
5×5	EDA	40.785±2.325	7.954
	DE/best/2/bin	42.167±3.284	2.714
	DE/rand/1/bin	39.851±1.730	2.810
	DE-EDA	39.292±0.012	3.767
10×10	EDA	34.785±1.907	121.44
	DE/best/2/bin	39.908±4.540	14.936
	DE/rand/1/bin	43.274±2.413	15.535
	DE-EDA	34.873±1.260	26.990

ACKNOWLEDGMENTS

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, and in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621063.

REFERENCES

[1] Bin Xin, Yangguang Zhu, Yulong Ding, and Guanqiang Gao. Coordinated motion planning of multiple robots in multi-point dynamic aggregation task. In *Proceeding of 2016 12th IEEE International Conference on Control and Automation (ICCA)*, pages 933–938. IEEE, 2016.
[2] Bin Xin, Guanqiang Gao, Yulong Ding, Yangguang Zhu, and Hao Fang. Distributed multi-robot motion planning for cooperative multi-area coverage. In *Proceeding of 2017 13th IEEE International Conference on Control & Automation (ICCA)*, pages 361–366. IEEE, 2017.
[3] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
[4] Pedro Larrañaga and Jose A Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2001.
[5] James C Bean. Genetic algorithms and random keys for sequencing and optimization. *Orsa Journal on Computing*, 6(2):154–160, 1994.