MFEA-IG: A Multi-Task Algorithm for Mobile Agents Path Planning

Yongjian Zhou, Tonghao Wang, Xingguang Peng*
School of Marine Science and Technology
Northwestern Polytechnical University
Xi'an, China
yjzhou416@gmail.com, neilwth@hotmail.com, pxg@nwpu.edu.cn

***Abstract*—Mobile agent path planning (MAPP) problem is a typical optimization problem. When we consider multiple agents path planning simultaneously, problems can be seen as multi-task optimization (MTO) problems. The Multi-factorial evolutionary algorithm (MFEA) is one promising technique for MTO problems. Within the MFEA some selective individuals that contain useful knowledge are transferred among independent tasks to enhance the convergence. In this work, we investigate what information, except to the selective individuals, should be transferred under the framework of the MFEA. In particular, the difference between the individuals and the estimated optimal solution of the corresponding task is used to calculate $individual\_gradient$ (IG), which is introduced into the MFEA as additional knowledge for transferring. Empirical studies on nine benchmarks validate the effectiveness of IG based MFEA (MFEA-IG). Moreover, we apply the MFEA-IG to MAPP problems. Simulation results show that the MFEA-IG outperforms the original MFEA and single task EA.**

***Index Terms*—multiple agents path planning, multi-task optimization, MFEA, knowledge transfer**

I. Introduction

Path planning is a fairly important technology for mobile agents. Existing path planning methods can be categorized into two categories [1]: classical and heuristic. Since the heuristic method is effective and efficient, it has attracted plenty of research interest. The researches that are part to the heuristic method include particle swarm optimization (PSO) [2], [3], ant colony optimization (ACO) [4], simulated annealing [5], etc. Genetic algorithm (GA) is one of the heuristic methods inspired by the process of natural selection. Due to its fast convergence, GA has been widely used to solve path planning problems. Farshchi et al. [6] and Karami et al. [7] applied GA to the dynamic environments for mobile robot path planning. Roberge et al. [8] used GA and PSO concurrently for the unmanned aerial vehicles (UAVs) path planning in a complex 3D environment, it shows that GA converges to the optimal solution faster than PSO.

Plenty of works have been done on path planning problems for single agent, but when multiple agents are considered, the problem becomes challenging. Wu et al. [9] proposed a GA-based algorithm to allocate multiple waypoints to multiple UAVs reasonably so that the total length of the path to be the shortest. In order to solve the problem of multiple agents colliding with obstacles or colliding with each other in an obstacle environment, Chakraborty et al. [10] designed a discrete motion planning frame, and use differential evolution (DE) algorithm to get the optimal position in next several time steps for multiple agents simultaneously. Trudeau et al. [11] proposed a decentralized local genetic programming (GP) approach to solve mobile agents path planning (MAPP) problems. This approach firstly studied an effective motion planning strategy for multiple agents using GP in a virtual environment. Then this strategy was deployed to each agent in the real environment. This GP-based approach shows robust scalability when the number of agents increases.

In this work, we investigate MAPP problems that can be solved in a multi-task manner. Specifically, MAPP is conducted for independent agents in a conjoint scene or for agents in separate scenes, which leads to the investigation of effective algorithms for multi-task optimization (MTO) problems. Multi-task Bayesian optimization [12] is a pioneer MTO paradigm algorithm. However, in recent years, evolutionary MTO (EMTO) has gained more attention. EMTO was first proposed in [13], which called multi-factorial evolutionary algorithm (MFEA). Based on evolutionary algorithm, implicit knowledge transfer technology is applied in MFEA to enhance the convergence of multiple tasks simultaneously. Then this framework was extended to multi-objective problems [14], [15]. When problems have different optimal location or their decision space have different dimensions, it is hard for MFEA to convergence to the optimal. Ding et al. [16] proposed a generalized multi-tasking algorithm based on MFEA to solve these two issues and successfully applied it to expensive problems. An explicit auto-encoding technology was used in MTO framework in [17]. Compared with MFEA, this algorithm could use multiple evolutionary search operators from different optimization algorithms.

In MTO framework, positive knowledge transfer among different tasks is the main motivation to improve the performance of the algorithm, and negative knowledge transfer [18] may lead to opposite results. In previous works, researchers pay more attention to decrease negative transfer (or increase positive transfer) [18], [19], make MTO more efficient with

This work has been supported by National Science Foundation of China (No. 61473233)

*Corresponding author

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

dynamic resource allocation strategy [20], etc. These algorithms are all evolutionary algorithm based, and the knowledge they transferred are all individual solutions, very few of which discuss what other information can be transferred. Inspired by this simple idea, we attempt to find a new kind of transferred knowledge in this paper. In addition to the individual solution, we use a kind of new knowledge, which is the difference between individual solution and the estimated optimal solution of its task as the additional knowledge to be transferred. And then, we apply our proposed algorithm to solve the MAPP problem.

The remainder of this paper is organized as follows. Section II describes a method to model the workspace of mobile agents and the evaluation function of the path. Then our proposed algorithm is introduced in Section III. Two experiments are designed to test the efficiency of our proposed algorithm in Section IV. Finally, we conclude our work and discuss some expectations of further research in Section V.

II. Problem Formulation

For mobile agents path planning problems, we use the same method to model each agent separately. Given a two dimensional environment with some stationary obstacles, path planning for mobile agents can be described as finding a set of waypoints between the start point and goal point that agent must pass through.

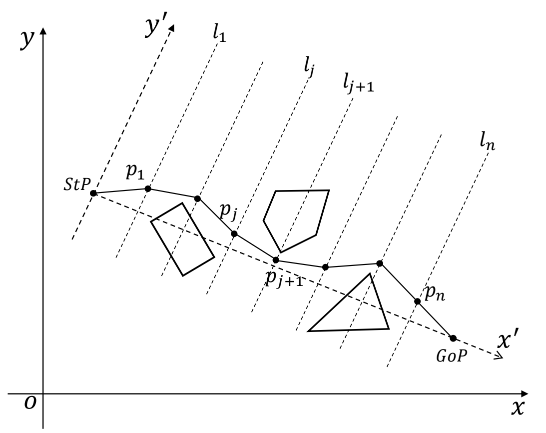


Fig. 1. Modeling method of a path.

Fig. 1 shows a simple yet effective method [21] to model the workspace of mobile agents. In the global coordinate, StP, GoP represent start point and goal point, respectively, and the polygon areas represent the obstacles. As can be seen from Fig. 1, a complete path can be expressed as

$$PATH = \{StP, p_1, p_2, ..., p_n, GoP\} \tag{1}$$

where $p_i (i = 1, 2, ..., n)$ is a waypoint of a path, and the path composed of these waypoints should be obstacle-free.

Thus, a suitable set of waypoints could represent a suitable path. Due to coordinates of waypoint p_i in xoy is a two dimensions data, it is pretty hard to find a suitable set of waypoints. Taking $StP-GoP$ as the x' axis, and the direction perpendicular to x' and passing through StP as the y' axis to build a rotate coordinate $x' - StP - y'$. Transformation method between two coordinate systems is as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \times \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} x_{StP} \\ y_{StP} \end{bmatrix} \tag{2}$$

α is the angle between axis x and x'. (x_{StP}, y_{StP}) is the coordinates of the start point StP in xoy.

Divide line $StP - GoP$ into $n + 1$ parts equally. $l_1, ..., l_n$ are lines obtained by making a vertical line at each bisection point. Waypoints $p_1, ...p_n$ are obtained on each vertical line $l_1, ..., l_n$. In coordinate system $x' - StP - y'$, the value of each waypoint in axis x' is constant. Then finding the suitable value of each waypoint in axis y' is same as finding a suitable path for agent. It is quiet easy to find the suitable path compared with it in xoy.

Length of the path can be calculated by the sum of the Euclidean distance of each two waypoints, and it can be described as follows:

$$L_{path} = L_{StP,p_1} + \sum_{i=1}^{n-1} L_{p_i,p_{i+1}} + L_{p_n,GoP} \tag{3}$$

$L_{p_i,p_{i+1}}$ is the length of two waypoints, it can be calculated by the follow equation:

$$L_{p_i,p_{i+1}} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{4}$$

(x_i, y_i) is the coordinate of point p_i in xoy. A suitable path for mobile agents should be obstacles free. For those paths which cross over the obstacles, a penalty should be part of the cost of paths:

$$Penalty = N_{obs} \times L_{long} \tag{5}$$

N_{obs} is the number of the path cross over obstacles. L_{long} is the longest side of the workspace for agents. Then the cost of path can be calculated by:

$$C_{path} = L_{path} + Penalty \tag{6}$$

Now we can get a better path for mobile agents by optimizing (6) .

III. Algorithm

A. Multi-factorial Evolutionary Algorithm

Multi-factorial evolutionary algorithm (MFEA) [13] is a MTO paradigm method to solve multiple tasks with one population simultaneously. By sharing useful knowledge among different tasks, MTO could have a better convergence to the optimal. Suppose that k tasks are to be optimizing, and, without loss of generality, the tasks are all single objective and to be minimizing. Then a MTO paradigm could be described as $\{x_1, x_2, ..., x_k\} = \arg\min(f_1(x), f_2(x), ..., f_k(x))$, $x_i = (x_i(1), x_i(2), ..., x_i(n))$ is the solution of ith task, n is the dimension of this task. With encoding all tasks to a $unfied\ space$ and decoding tasks to their specific space when evaluating them, MFEA could have $implicit\ knowledge\ transfer$ [17] during its optimizing process. To evaluate each individual in the population, some properties defined in MFEA are as follows:

Definition 1 (factorial_cost): Factorial cost is the evaluation of all k tasks corresponding to an individual ind_i, each dimension represents the fitness of a task, so it is a k dimensions vector.

Definition 2 (factorial_rank): Factorial rank r_{ij} is the index of individual ind_i in the population sorted in ascending order with respect to their factorial costs on task j.

Definition 3 (skill_factor): Skill factor τ_i of individual ind_i is the index of the task whose factorial rank is smallest among all tasks.

Definition 4 (scalar_fitness): Scalar fitness φ_i of individual ind_i is calculated by $\varphi_i = 1/\tau_i$.

B. MFEA-IG

Though many researchers focus on promoting the performance of EMTO, very few have studied what other knowledge but the individual solution can be transferred among different tasks. In this section, we use MFEA as the basic algorithm and discuss a new kind of knowledge named $individual\_gradient$ as the additional transferred knowledge.

Applying useful knowledge transfer among different tasks, EMTO can get better performance than single task optimization. Considering this useful knowledge as reflecting some characteristics of the original task, there must be other knowledge we could use besides the individual solution. We utilize the estimate optimal solution x^i_{esti} of the i-th task and individual solution x^i_k to calculate $individual\_gradient$.

Definition 5 (individual_gradient): Individual gradient $individual\_grad_k$ is the distance with direction between individual k and the estimate optimal solution with respect to task τ_i

Algorithm 1 shows the structure of our proposed algorithm $indiviudal\_gradient$ based MFEA (MFEA-IG). Following the basic algorithm MFEA, we first initialized N $individuals$ randomly in the unified space [0,1] and named them as $parent\_pop$. From lines 2 to 5, we assign the skill factor to each individual and evaluate them with respect to the corresponding task. Line 8-30 shows the main process of MFEA-IG. Every GEN_TRANS generations we use our proposed method to compute the $individual\_grad_i$ of each individual ind_i, which is introduced in algorithm 2. It is the same as the original MFEA from line 12 to 29. The basic GA operators $crossover$ and $inherited$ are used to generate the $offspring\ population$ in a multitask paradigm. Implicit knowledge transfer occurs during these process. We use $NormalDistribution$ as the basic mutate operator, and set $individual\_grad_i$ instead of 0 as the mean value for the distribution to utilize this $knowledge$. Then we combine the $offspring\ population$ and $parent\ population$ to form the $inter\ population$ and use a selection method to select N fittest individuals to form the new $parent\ population$.

Algorithm 2 shows the detailed process of calculating $individual\_gradient$. We estimate the optimal

Algorithm 1 Pseudo-code of the MFEA-IG

1: Randomly initialize N individuals and store them to $parent\_pop$
2: **for** each individual ind_i in $parent\_pop$ **do**
3: Assign skill factor τ_i=$mod(i, K) + 1$, for the case of K tasks.
4: Only evaluate ind_i for task τ_i .
5: **end for**
6: Set gen = 1
7: Set GEN_TRANS
8: **while** $gen\ <\ GENERATION$ **do**
9: **if** gen % GEN_TRANS == 0 **then**
10: compute $individual\_grad_i$ of each individual ind_i
11: **end if**
12: $offspring\_pop$=ϕ
13: **for** i=1:[N/2] **do**
14: Pick two individual $ind_i, ind_{N/2+i}$ from $parent\_pop$
15: **if** rand(0,1)< $rmp\ or\ \tau_i == \tau_{N/2+1}$ **then**
16: $o_1, o_2 = Crossover(ind_i, ind_{N/2+i})$
17: Assign offspring o_1, o_2 skill factor $\tau_i\ or\ \tau_{N/2+1}$ randomly
18: **else**
19: o_1=$NormalDistribution(grad_i, sigma)$
20: o_2=$NormalDistribution(grad_{N/2+i}, sigma)$
21: $\tau_{o_1}, \tau_{o_1} = \tau_i, \tau_{N/2+1}$
22: **end if**
23: $offspring\_pop = offspring\_pop \bigcup [o_1, o_2]$
24: **end for**
25: Evaluate each individual in $offspring\_pop$ only for their assigned skill factors.
26: $inter\_pop = offspring\_pop \bigcup parent\_pop$
27: Update the scalar fitness and skill factor of every individual in $inter\_pop$
28: Select the fittest N individuals from $inter\_pop$ to form the new $parent\_pop$
29: gen=gen+1
30: **end while**

$EstiOptimal_i$ of the i-th task by the following equation,

$$EstiOptimal_i = \sum_{k=1}^{EstiNum} Fittest^i_k / EstiNum \qquad (7)$$

$EstiNum$ is the number of individuals we use to estimate the optimal of tasks. $Fittest^i$ is the fittest solution for the i-th task sorted by ascending order. We use $shifting\_rate$ in the range of [0,1] to degree of $individual\_grad_k$.

IV. Simulation Analysis

In this section, we first evaluate the performance of our proposed algorithm compared with the original MFEA both on nine single objective MTO benchmarks [22]. Similarity and intersection are used to described the characteristics of these benchmarks to verify how these characteristics affect the effectiveness of the algorithm. Then we applied MFEA-IG to MAPP and compared with the MFEA and single

Algorithm 2 Process of calculating $individual\_gradient$

Input: $parent\_pop, Tasks$
Output: $updated\ parent\_pop$

$num\_of\_task = length(Tasks)$
Set $EstiNum$
Set $shifting\_rate$
for i=[1:num_of_task] **do**
  Calculate the estimated optimal $EstiOptimal_i$ of the i-th task according to Equation 7
  for each ind_k in $parent_pop$ **do**
    if $skill\_factor$ of ind_k is i **then**
      $individual\_grad_k$=$shifting\_rate$*($EstiOptimal_i$-x_k)
    end if
  end for
end for

task evolutionary algorithm (STEA). SBX for crossover and Gaussian mutation are employed to the three algorithms.

A. Empirical studies on benchmark functions

We first compare our algorithm with the original MFEA on nine benchmarks. The parameters setting for two algorithms are as follows: the population size N=50, the number of iteration $GENERATION = 100$, the parameter rmp is set as 0.3. The parameter $EstiNum$ is set as 5 and $shifting\_rate$ is set as 0.3 for MFEA-IG. The number of independent runs is set as 20. Table I shows the mean value of the final generation of 20 runs for each task. We use the Wilcoxon rank sum test with 95% confidence level to check the performance of two algorithms, and the better results are the highlight in bold. As can be seen from Table I, MFEA-IG performs better than MFEA on 5 out of 18 tasks. It is clear that the use of additional knowledge can improve the effectiveness of MFEA-IG.

TABLE I
Performance on nine benchmark functions

Benchmark	Task	MFEA	MFEA-IG
f1	Griewank	0	0
	Rastrigin	0	0
f2	Ackley	4.92	4.75
	Rastrigin	126.84	106.31
f3	Ackley	19.99	**19.99**
	Schwefel	6499.86	6190.88
f4	Rastrigin	310.35	283.27
	Sphere	5.25E-13	5.19E-13
f5	Ackley	1.68	1.34
	Rosenbrock	23.12	27.73
f6	Ackley	18.16	**11.63**
	Weierstrass	24.41	**14.69**
f7	Rosenbrock	33.79	**17.23**
	Rastrigin	3.63	**0**
f8	Griewank	1.11E-11	1.46E-11
	Weierstrass	29.51	29.31
f9	Rastrigin	420.36	401.35
	Schwefel	6859.64	6901.24

B. Applying to multiple agents path planning

In this section, we apply our proposed MFEA-IG to MAPP and compared it with the classical MFEA and single task evolutionary algorithm (STEA). We use the cost of path C_{path} as the fitness of the task and use the same parameters in the following experiments: the population size $N = 30$, the path is divided into 31 parts, which means the tasks are 30 dimensional, the number of iteration $GENERATION = 30$. The parameter rmp is set as 0.3 for MFEA and MFEA-IG, parameter $EstiNum$ is set as 5 for MFEA-IG and parameter $shifting\_rate$ is set as 0.3 for MFEA-IG. The number of independent runs is set as 20.

Different agents can work in the different workspace. Assuming their workspaces have the same size of 100×100. The first experiment tests two agents work in the same workspace, and the second tests in the different workspace. Then we select the median result of 20 independent runs from each algorithm and display them in Fig. 2 and Fig. 3. Table II and Table III show the mean value of the final generation of 20 runs for each algorithm.

(1) **Experiment 1**

This experiment set multiple agents to work in the same workspace, which includes 7 polygon obstacles. Assuming the start and goal positions of the first agent are (7.84,90.56), (76.58,40.12), and the second agent are (9.76,70), (75.11,20.53).

TABLE II
Performance of three algorithms on Experiment 1

	Mean Values of Path Length		
	MFEA-IG	***MFEA***	***STEA***
path1	**128.37**	143.68	154.98
path2	**116.48**	132.68	136.82

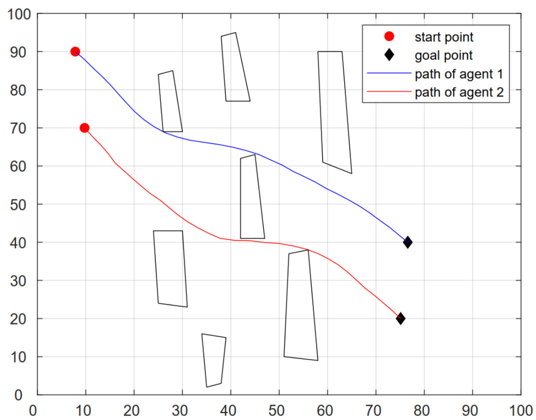
(2) **Experiment 2**

The experiment 2 set multiple agents work in the different workspace, which the first is the same as experiment 1, and the second includes 5 polygon obstacles. Assuming the start and goal positions of the first agent are (7.84,68.45), (76.58,70), and the second agent are (7.65,44.45), (81.07,32.52). The coordinates of the second workspace obstacles are as follows:

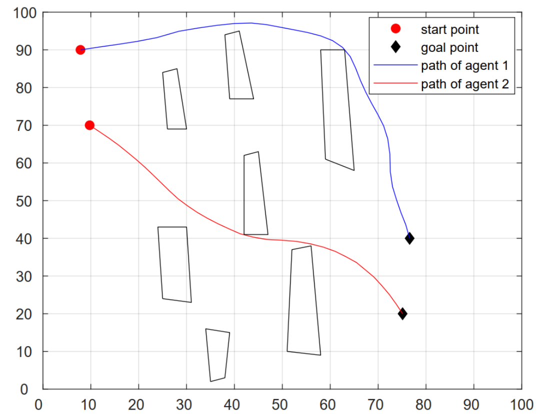
TABLE III
Performance of three algorithms on Experiment 2

	Mean Values of Path Length		
	MFEA-IG	***MFEA***	***STEA***
path1	**122.91**	137.38	154.03
path2	**92.38**	104.07	111.15

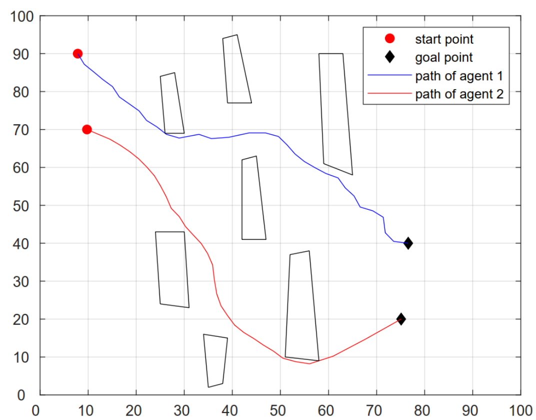
Table II and III demonstrate that MFEA-IG performs better than the other two algorithms, which confirm that use $individual\_gradient$ as the additional transfer knowledge in MFEA framework is capable of finding a better solution. The



(a) MAPP based on MFEA-IG



(b) MAPP based on MFEA



(c) MAPP based on STEA

Fig. 2. MAPP in one workspace.

same conclusion also can be obtained from Fig. 2 and Fig. 3, what they show is the median of 20 results, but the results obtained by MFEA-IG are much better than those of the other two algorithms.

V. Conclusion

In order to pursue more possibilities of knowledge transfer among different tasks, in this paper, we propose MFEA-IG. Rather than sharing regular individual solutions in conventional approaches, we introduce $individual\_gradient$ into MTO framework as the additional knowledge to be transferred. Empirical studies on nine benchmarks showed that MFEA-IG performs better than MFEA on 5 out of 18 tasks. Then we apply this algorithm to several MAPP problems. Two simulations involving multiple agents working in the same workspace and in the different workspaces are designed to test our algorithm. The results demonstrate that the proposed algorithm is effective and efficient in MAPP problem. In the future, we will discuss what other knowledge can be transferred and how to use the knowledge more effectively and efficiently in MTO paradigm. Also, we will try to apply our algorithm to MAPP in dynamic environments.

References

[1] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2016.08.001

[2] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, 2013.

[3] J. Tang, J. Zhu, and Z. Sun, "A novel path planning approach based on appart and particle swarm optimization," in *International Symposium on Neural Networks*. Springer, 2005, pp. 253–258.

[4] X. Chen, Y. Kong, X. Fang, and Q. Wu, "A fast two-stage ACO algorithm for robotic path planning," *Neural Computing and Applications*, vol. 22, no. 2, pp. 313–319, 2013.

[5] Q. Zhu, Y. Yan, and Z. Xing, "Robot path planning based on artificial potential field approach with simulated annealing," *Proceedings - ISDA 2006: Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 622–627, 2006.

[6] S. M. R. Farshchi, S. A. NezhadHoseini, and F. Mohammadi, "A Novel Implementation of G-Fuzzy Logic Controller Algorithm on Mobile Robot Motion Planning Problem," *Computer and Information Science*, vol. 4, no. 2, pp. 102–114, 2011.

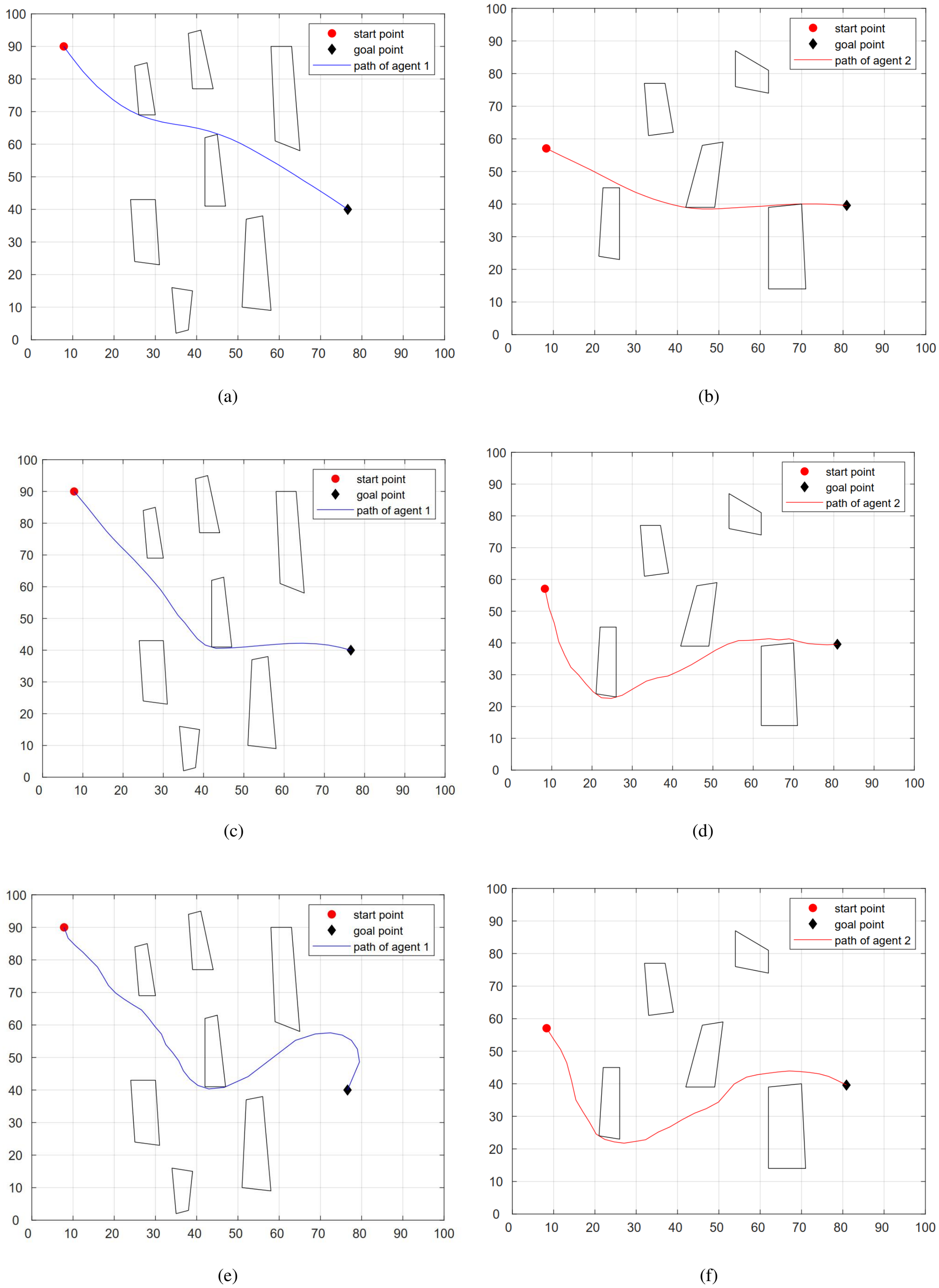


Fig. 3. MAPP in different workspaces. (a) and (b) are the results based on MFEA-IG. (c) and (d) are the results based on MFEA. (e) and (f) are the results based on STEA.

[7] A. H. Karami and M. Hasanzadeh, "An adaptive genetic algorithm for robot motion planning in 2D complex environments," *Computers and Electrical Engineering*, vol. 43, pp. 317–329, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2014.12.014

[8] V. Wu2007, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[9] X. Wu, Z. Feng, J. Zhu, and R. Allen, "GA-based path planning for multiple AUVs," *International Journal of Control*, vol. 80, no. 7, pp. 1180–1185, 2007.

[10] J. Chakraborty, A. Konar, L. C. Jain, and U. K. Chakraborty, "Cooperative multi-robot path planning using differential evolution," *Journal of Intelligent and Fuzzy Systems*, vol. 20, no. 1-2, pp. 13–27, 2009.

[11] A. Trudeau and C. M. Clark, "Multi-Robot Path Planning Via Genetic Programming," pp. 1–18, 2019. [Online]. Available: http://arxiv.org/abs/1912.09503

[12] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task bayesian optimization," in *Advances in neural information processing systems*, 2013, pp. 2004–2012.

[13] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial Evolution: Toward Evolutionary Multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2016.

[14] B. Da, A. Gupta, Y.-S. Ong, and L. Feng, "Evolutionary multitasking across single and multi-objective formulations for improved problem solving," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 1695–1701.

[15] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE transactions on cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.

[16] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized Multitasking for Evolutionary Optimization of Expensive Problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 44–58, 2019.

[17] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y. S. Ong, K. C. Tan, and A. K. Qin, "Evolutionary Multitasking via Explicit Autoencoding," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3457–3470, 2019.

[18] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii," *IEEE Transactions on Evolutionary Computation*, vol. 24, pp. 69–83, 2019.

[19] X. Zheng, A. Qin, M. Gong, and D. Zhou, "Self-regulated evolutionary multi-task optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, pp. 16–28, 2019.

[20] M. Gong, Z. Tang, H. Li, and J. Zhang, "Evolutionary multitasking with dynamic resource allocating strategy," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 858–869, 2019.

[21] B. Sun, W.-d. Chen, and Y.-g. Xi, "Particle swarm optimization based global path planning for mobile robots," *Control and Decision*, vol. 20, no. 9, p. 1052, 2005.

[22] B. Da, Y.-S. Ong, L. Feng, A. K. Qin, A. Gupta, Z. Zhu, C.-K. Ting, K. Tang, and X. Yao, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," *arXiv preprint arXiv:1706.03470*, 2017.