A destination swap scheme for multi-agent system with agent-robots in region search problems

Zonglin Ye, Shuo Yang, Shuo Zhu, Yazhe Tang,* Hui Cao,† and Yanbin Zhang ‡

ABSTRACT

As the multi-agent robots can share information and cooperate with each other, they are applied in the region search problems like odor source localization. In this kind of problems, the detection result of the agent-robot at the current destination will influence the location of its next destination. So the route planning of the agent-robots cannot be made in advance. This paper proposes a dynamic destination allocation scheme for the agent-robot group to reduce the total moving distance of the agent-robot group and thus save energy. When the agent-robots start to search the target region, the scheme calculates the first destinations for the agent-robots and allocates the destinations to the agent group to minimize their total moving distance. Then as the distances between the agent-robots and their destinations are different, the agent-robots will reach their target destinations at different moments. In the following searching process, whenever an agent-robot arrives at its current destination and calculates its next destination, the scheme will then find out the best destination allocation plan for the agent-group to minimize the total moving distance of the group. The simulation is performed with particle swarm optimization (PSO) on benchmark functions with agent-robots. The experiments verify the effectiveness of the proposed scheme.

1 INTRODUCTION

The multi-agent system (MAS) consists of a number of agents which can interact with each other and react to the environment. Due to its superiority in dealing with complex problems, the notion of MAS has been applied in many aspects like commercial bargain modelling [1], optimization problems [2], wireless sensor network (WSN) [3], and distributed surveillance system [4].

In this kind of application, the agent is a programme sent by the sink node in the WSN or the central computer in the distributed surveillance system when a mission is generated. That is to say, the agents in the application above are virtual agents, not entities. They do not occupy any space, will not be worn, and cost nearly no energy.

However, for multi-agent robots, like robots used for region search operations [5] (e.g. odor source localization [6]), the situation is different. In these two applications, agent-robots consume energy while moving. So the designer should not only make sure that the problem can be solved, but also reduce the energy consumption. Decreasing the total moving distance of the agent-robot group is an effective approach and is meaningful for all the group-search problems with agent-robots.

There are many works focus on the task allocation problems of the agent-robot group or the UAV group [7, 8]. However, in these works, the locations of all the targets are known before the allocation. As in the region search problems like odor source localization, the locations of the targets to be detected are calculated real-timely with the information obtained from the former detection results, other methods are needed to find an optimal allocation scheme for the region search problems.

This paper proposes a dynamic destination allocation scheme for multi-agent system with agent-robots in the region search problems to reduce the total moving distance of the agent-robot group. When the agent-robots start to search the target region, the scheme first calculates the first destinations under the predefined searching algorithm for the agent-robots and allocates the destinations to the agent group to minimize their total moving distance. Then as the distances between the agent-robots and their destinations are different, the agent-robots will reach their target destinations at different moments. In the following searching process, whenever an agent-robot arrives at its current destination and calculates its next destination, the scheme will then find out the best destination allocation plan for the agent-group to minimize the total moving distance of the group. Generally speaking, the destination re-allocation scheme is triggered whenever an agent-robot reaches calculates its next destination. As particle swarm optimization (PSO) [9] is widely used in the region search problems, a simulation of 2-D PSO benchmark function optimization problem with agent-robots using PSO is performed to show the effectiveness of the proposed scheme. As the conventional PSO is a synchronous algorithm and the agent-robots will reach its distance and different moment, an asynchronous version of PSO is used. In this asynchronous PSO, the particles are treated as the real-world robots and

*Yazhe Tang is with the Temasek Laboratories, National University of Singapore, Singapore (email: yztang2008@yahoo.com)

†Corresponding author. Tel.: + 86 13991193207; fax: + 86 029-82668666-2113. Email address: huicao@mail.xjtu.edu.cn

‡Zonglin Ye, Shuo Yang, Shuo Zhu, Hui Cao and Yanbin Zhang are with State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xian Jiaotong University, Xian, Shaanxi, 710049

IMAV 2016, Beijing, PR of China, 17-21 October 2016
International Micro Air Vehicle Competition and Conference 2016

their movements cost time. As the distances between the locations of the particles and their destinations are different and we assume that the velocities of the particles are identical, the particles will arrive at their destination at different moments. Whenever a particle arrives at its current destination, its next destination is calculated by its personal best and the current global best. Thus, in this version of PSO, the different particles will have different update times at a specific moment. The experiments verify the effectiveness of the proposed scheme. The influence of the number of the agent-robot to the scheme is also analysed.

The rest of the paper is organized as follows. Section II is the proposed scheme. Section III is the experiment and section IV gives the conclusion.

2 THE PROPOSED SCHEME

In region search problems, the agent-robots need to move in the searching region to find the problem-related target under the predefined algorithm. No matter what the problem and the algorithm are, the agent-robots are always in the iteration composed of three parts: detecting the destination, calculating the next destination, and moving to the next destination. The detection part is accomplished by the sensors equipped on the agent-robot. Then the agents can exchange the information of the problem, like the optimal value being detected and the locations of the obstacles. When calculating the next destination for every agent, the agent-robots are treated as a group in the sense that the knowledge gathered by all the agent-robots is used. However, when moving to the next destination, the agent-robot just moves to the destination calculated by itself. In another word, in the moving process, the agent-robots are treated as individuals and there is no cooperation among them. The following part introduces a dynamic destination allocation scheme for the agent-robot group.

The destination allocation scheme is based on the fact that the next destination of an agent-robot may be closer to another agent-robot. In that situation, if the two destinations are re-allocated between the two agent-robots, the total moving distance and the energy consumption of the two agent-robots can be decreased. In the scheme, the parts of detecting the destination and calculating the next destination are the same as the conventional scheme in the region search problems. Now we show the proposed destination allocation scheme for the process of moving to the next destination. Assume that there are m agent-robots $r_1, r_2, ..., r_m$ in the searching region to detect the target which is at a specific location *X*. The searching process is controlled by a specific algorithm *C*. The process of the scheme can be described as follows:

Step 1: The calculation of the initial locations and the first destinations of the agent-robots. The initial locations of the agent-robots, donated as $x_1^0, x_2^0, x_3^0, ..., x_m^0$ are obtained by *C*. The subscript is the label of the agent-robot and the superscript is the iterations of the agent-robot. For example, x_3^5 is the third location of the 5th agent-robot. The second location of agent-robot *i*, donated as x_i^1 , is also obtained by *C*.

Step 2: The first destination allocation process of the agent-robot group. The purpose of this step is to minimize the total moving distance of the robot group. At the beginning, the agent-robots are at $x_1^0, x_2^0, x_3^0, ..., x_m^0$, and their destinations are $x_1^1, x_2^1, x_3^1, ..., x_m^1$. In conventional algorithms without considering the cooperation in the moving process, the agent-robots will move to their destination, respectively. And the total moving distance of the robot group is $\sum dist(x_i^1 - x_i^0)$, where *dist* is the Euclidean distance. The destination allocation process is used to minimize the total moving distance.

For example, if there are agent-robots *i*, *j*, and *k* with $x_i^1 = p_i$, $x_j^1 = p_j$, $x_k^1 = p_k$, where p_i, p_j, and p_k are the locations of the destinations after step 1. Then after Hungarian algorithm gives an allocation plan, the allocated destinations may be like $x_i^1 = p_i$, $x_j^1 = p_k$, $x_k^1 = p_j$. In this case, the agent-robots *j* and *k* have exchanged their destinations to decrease the total moving distance.

Step 3: The agent-robots with the least moving distance moves to its current destination and gets its new destination.

For an agent-robot *i*, after step 2, its moving distance will be $dist(x_i^1 - x_i^0)$. For the agent-robot group, there will be a robot, labeled as least, with the least moving distance. As we assume that the velocity of all the agent-robots are 1, then robot least will first reach its destination x_{least}^1 after $t_1 = dist(x_i^1 - x_i^0)$. Here t_1 is the time interval between the first and the second allocation. Then based on the information obtained from location x_{least}^1 , the predefined searching algorithm *C* calculates the next location of the agent-robot *least*, which is donated as x_{least}^2 .

Step 4: The destination allocation process of the agent-robot group. When the next destination of the agent with the least moving distance, $x_{least}^{t_{least}+1}$ is obtained, another destination allocation for the agent-robot group is performed. The *l*th destination allocation can be formulated as follows:

$$\arg\min(\sum_{i=1}^{m} dist(x_i^1 - k_{ij}x_j^0))$$

$$s.t. \begin{cases} \sum_{i=1}^{m} k_{ij} = 1 & j = 1,2,3,...,m \\ \sum_{j=1}^{m} k_{ij} = 1 & i = 1,2,3,...,m \\ k_{ij} = 1 \, or \, 0 & i,j = 1,2,3,...,m \end{cases} \tag{1}$$

where x_{jc}^{nj} is the current location of agent-robot *j* when the *l*th destination allocation happens. n_j is the number of times that agent-robot *j* is the first to reach its destination, which also can be interpreted as the times that agent-robot *j* gets its destination by *C*, not by the allocation process. This is a combinatorial optimization problem and can be solved by

Hungarian algorithm which solves the assignment problem in polynomial time [10]. The algorithm can give an allocation plan to the agent-robot group.

$$x_{jc}^{n_j} = x_j^{n_j} + \frac{x_j^{n_j+1} - x_j^{n_j}}{\|x_j^{n_j+1} - x_j^{n_j}\|} \cdot \|x_{least}^{n_{least}+1} - x_{least}^{n_{least}}\| \quad (2)$$

$$x_{least\,c}^{n_{least}} = x_{least}^{n_{least}+1} \quad (3)$$

where $x_{least}^{n_{least}+1}$ least is updated by *C*.

In our scheme, Hungarian algorithm is also used to solve (2) and gives a destination re-allocation plan for the agent-robot group.

Step 5: Iterate step 3 and 4, until algorithm *C* ends.

The following section will illustrate the performance of the proposed scheme on some benchmark functions of PSO algorithm, which is widely used in region search problems.

3 Experiments

In this section, six test functions (Bohachevsky, Griewank, Michalewicz, Rastrigin, Rosenbrock, and Schwefel) are used to verify the effectiveness of the proposed scheme. The concept of PSO was first suggested by Kennedy and Eberhart [9]. The algorithm uses a group of particles to search the feasible region and updates the locations of the particles by considering the current location of the particle, the history best location of the particle, and the global best location of the particle group. For the *i*th particle P_i, let v_{id}^t donate the velocity on the *d*th dimension at the *t*th iteration. Similarly, x_{id}^t is its location on the *d*th dimension at the *t*th iteration. Let p_{id}^t and p_{gd}^t be the particle best location and swarm best location till the *t*th iteration. Then the velocity of the *i*th particle on the *d*th dimension is updated according to the following equation [9]:

$$v_{id}^t = wv_{id}^{t-1} + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (4)$$

where $d = 1, 2, ..., D$, and D is the dimension of the searching space. $c_1$and c_2 are acceleration factors indicating the degree that the particle is attracted by its personal best position and the global best position. These two constants are set as 2 in order to make the average velocity change coefficient close to 1 [9]. r_1 and r_2 are uniformly generated random numbers with a scope of [0, 1]. w is the weighting factor controlling the searching region of the particles. Then the location of the *i*th particle on the *d*th dimension is updated as follows:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t \quad (5)$$

For the experiments with real robots, moving from one place to another costs time. Usually, the agent-robots will not arrive at their destinations at the same time. Comparing with the strategy that updating the new destinations synchronously when each member of the agent-robot group reaches its destination, we would rather do the destination allocation process whenever an agent-robot reaches the destination. Thus, unlike the conventional PSO, the PSO used in the experiment is an asynchronous PSO. The asynchronous PSO is

$$v_i^{n_i} = wv_i^{n_i-1} + c_1 r_1 (p_i^{n_i} - x_i^{n_i}) + c_2 r_2 (p_g^l - x_i^{n_i}) \quad (6)$$

$$x_{id}^{n+1} = x_{id}^n + v_{id}^n \quad (7)$$

The experiment uses the asynchronous PSO described above and the asynchronous PSO with the destination allocation scheme on six benchmark functions to compare the total moving distance of the agent-robot group in the entire searching process. For each function, the asynchronous PSO algorithm and the asynchronous PSO with the proposed scheme both run for 100 times. The number of the iterations of the algorithm is set to be 1000 before it ends in each run. The average value of the total moving distance of the agent-robot group in the two methods, the percentage of the distance decreased by the proposed scheme, and the results of the Wilcoxon rank-sum test are listed in table 1. From Table 1, it is clear that the proposed scheme can significantly reduce the moving distance of the agent-robot group for all the benchmark problems. For all the functions, the proposed scheme helps to decrease the total moving distance of the agent-robot group by nearly one quarter except one function. The maximum reduction happens on the Rastrigin function, which is 24.2 %. With a significance level of 5%, the p-value also indicates that the superiority of the proposed scheme does not happen by chance. And the superiority also holds over Bohachevsky function.

Another experiment is performed to investigate the relationship between the number of the robots and its effect on the proposed scheme. For Griewank function, the experiment is performed for 100 times with the number of the robots increasing from 6 to 12. The average values of the total moving distance of the two schemes in 100 experiments of Griewank function with the number of the robots ranging from 6 to 12 are in Table 2. From this table, it can be found that the total moving distance of the two schemes change with the number of the robots. The total moving distance of the proposed scheme is always smaller than that of the conventional PSO. The percentage of distance decreased varies, ranging from 5.7% to 47.4%. It can be seen that the number of the robots is not the only element that affects the total moving distance. More experiments are needed to discover the other elements.

4 Conclusions

This paper proposes a dynamic destination allocation scheme for the agent-robot group to reduce the total moving distance of the agent-robot group and thus save energy. As PSO is widely used in region search problems, experiments

Benchmark functions	Asynchronous PSO with proposed scheme	Asynchronous PSO	Decreased percentage (%)	P-value
Bohachevsky	3.0040E+03	3.1384E+03	4.4	4.9304E-04
Griewank	1.3593E+04	1.7271E+04	22.3	1.6611E-16
Michalewicz	682.0137	883.4735	22.8	1.4568E-18
Rastrigin	1.0352E+04	1.3666E+04	24.2	1.5763E-20
Rosenbrock	9.8706E+03	1.2909E+04	23.6	3.2185E-19
Schwefel	3.9774E+05	5.2008E+05	23.5	1.9068E-15

Table 1: The average of the total moving distance on six benchmark functions of the two schemes

The number of the robots	The proposed scheme (1E+03)	Asynchronous PSO (1E+03)	Decreased percentage (%)
6	0.6375	0.7141	10.7
7	0.8610	0.9892	13.0
8	1.3593	1.7271	21.3
9	1.2319	1.5703	21.6
10	1.6122	2.0451	21.1
11	1.6041	3.0515	47.4
12	1.6042	1.7009	5.7

Table 2: The total moving distance with the changing of the number of the robots for the two schemes

are performed on the benchmark functions to verify the effectiveness of the proposed scheme. From our research, several conclusions can be obtained. First, the proposed scheme can significantly reduce the total moving distance of the agent-robot group. Second, like the conventional PSO, the asynchronous PSO can find the optimal value of the region search problem. Finally, the asynchronous PSO with the proposed scheme can reduce the total moving distance of the agent-robot group compared to the asynchronous PSO itself. In the future, the simulations and experiments on practical region search problems like odor source localization need to be implemented. More properties of the asynchronous PSO also deserve further analysis.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant 61375055, the Program for New Century Excellent Talents in University under Grant NCET-12-0447, the Natural Science Foundation of Shaanxi Province of China under Grant 2014JQ8365, State Key Laboratory of Electrical Insulation and Power Equipment EIPE16313 and the Fundamental Research Funds for the Central University.

REFERENCES

[1] M.H. Chen and L. Wang. Impact of individual response strategy on the spatial public goods game within mobile agents. *Applied Mathematics and Computation*, 251:192–202, 2015.

[2] X.F. Xie and J.M. Liu. Multiagent optimization system for solving the traveling salesman problem (tsp). *IEEE Transactions on Systems Man And Cybernetics Part B*, 39(2):489–502, 2009.

[3] D.Y. Ye, M.J. Zhang, and D. Sutanto. Self-organization in an agent network: A mechanism and a potential application. *Decision Support Systems*, 53(3):406–417, 2012.

[4] J.M. Gascuena and F.C. Antonio. On the use of agent technology in intelligent, multisensory and distributed surveillance. *The Knowledge Engineering Review*, 26(2):191–208, 2011.

[5] D.W.Gong and C.L. Qi. Modified particle swarm optimization for odor source localization of multi-robot. In *IEEE Congress on Evolutionary Computation (CEC)*, 2011.

[6] A.T. Hayes, A. Martinoli, and R.M. Goodman. Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, 21(4):427–441, 2003.

[7] M. Alighanbari, Y. Kuwata, and J.P. How. Coordination and control of multiple uavs with timing constraints and loitering. In *Proceeding of the IEEE American Control Conference*, 2003.

[8] S. Rathinam and R. Sengupta. Lower and upper bounds for a multiple depot uav routing problem. In *Proceeding of the IEEE American Control Conference*, 2003.

[9] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, 1995.

[10] Pentico and W. David. Assignment problems: A golden anniversary survey. *European Journal Of Operational Research*, 176(2):774–793, 2007.