Proceedings of the 2014 IEEE
International Conference on Robotics and Biomimetics
December 5-10, 2014, Bali, Indonesia

The Multi-target Search Problem with Environmental Restrictions in Swarm Robotics

Jie Li, Ying Tan

***Abstract*— In this paper, we give a comprehensive and detailed introduction of the multi-target search problem in swarm robotics. Based on some assumptions, we built the idealized models of the basic search problem and three kinds of environmental restrictions. In our previous works, we have raised two searching strategies (GES and IGES) inspired from firework explosion and three restriction-handling strategies, and in this paper, we describe and analyze these strategies systematically. A series of experiments were carried out, and the results show that the strategies proposed work well on the idealized models. And it's valuable to note that, compared to GES and RPSO, IGES was more efficient and showed greater stability in searching process, greater adaptiveness in both small and large scale problems and greater compatibility with restriction-handling strategies.**

I. INTRODUCTION

Swarm robotics has achieved great progress because of the development of artificial intelligence [1]. Swarm robotics can find its position in many applications, and it's especially suitable for tasks requiring large amounts of individuals, and for operations difficult or dangerous for human beings, e.g. foraging [2], surveillance [3], monitoring [4] and search-and-rescue [5]. These applications can be abstracted as a multi-target searching problem. Multi-target search in swarm robotics is a process in which a swarm of robots try to find and collect large amounts of targets distributed in the vast unknown environment.

In search stage, robots can perceive vague information about targets, such as the approximate distance between targets and current position. The vague information can be regard as fitness values, which have corresponding meanings in physical world, such as Euclidean distance [6], olfaction measurements [7] or chemical clues [8] and potential functions [9]. These fitness values are continuous and such problems can be solved with gradient decent methods [10] or other local searching schemes [11]. However, hardware designs in swarm robotics should be as simple as possible which may leads to low quality on-board sensors and fault sensing results and errors [12]. To make the problem more realistic and challenging, discrete fitness values and various environmental restrictions are introduced into the problem.

Thanks to the similarity in problem, many swarm intelligence algorithms and their variants are used as the cooperative strategy of swarm robots, such as PSO [13], ACO [14] and firework algorithm [15]. In this paper, we choose three underlaying searching algorithms to cope with various restriction-handling strategies. One algorithm is RPSO [16], the other two algorithms are GES [17] and IGES [18], proposed in our previous work, and our paper [19] proposed strategies for avoiding decoys in the problem.

The problem of multi-target search in swarm robotics is stated in Section II. And Section III introduces the environmental restrictions to be tackled. Then Section IV presents two searching strategies we proposed. And the strategies for handling restrictions are introduced in Section V. Experimental results and discussions are presented in Section VI. Finally, Section VII concludes our paper.

II. PROBLEM STATEMENT

To define precisely the problem of multi-target search in swarm robotics, we made some assumptions, on which we constructed an model.

A. Assumptions

- Environment: a number of targets, a swarm of robots.
- Targets: remain stationary, can be removed from the environment, may obey some distribution, generate positive fitness values around them (the farther, the smaller).
- Scopes of influence: each target has its own scope of influence; may obey some distribution, the fitness value of overlapped area is not less than that generated by each target at that area.
- Robots: can be added into or removed from the environment; have no prior knowledge of the number, the distribution and the influence scopes of targets.
- Abilities of each robot: local perception, local interaction, simple mobility (with a maximum speed limit), simple decision-making capacity, limited memory.
- Three kinds of perception: fitness value perception, object perception, target perception.
- Interaction: limited to a certain range; obtain fitness values of neighbor robots.
- Robots swarm: no leader, no central control, no uniform number, starts from one region; all robots have the same hardware and software, and each robot makes decisions according to obtained information and executes them independently; cooperation of multiple robots can accelerate the target collection.
- Iteration: each robot executes a series of actions at one iteration; in order to exploit meticulously and avoid

All authors are with Key Laboratory of Machine Perception and Intelligence (Peking University), Ministry of Education; Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, China, 100871.

This work was supported by the National Natural Science Foundation of China with grants no. 61375119, 61170057 and 60875080. Professor Ying Tan is the corresponding author.ytan@pku.edu.cn

missing the optimum position, the difference of fitness values in two adjacent iterations should be small.
- Evaluation criteria: time cost, performance under environmental restrictions.

B. An Idealized Model

An idealized model of the multi-target search problem is shown in Figure 1, which is described as following:

- Environment: abstracted to be a square (side length is 1000 units), holds m targets and n robots.
- Each robot: abstracted to be a square (side length is 1 unit), can memorize information of 10 iterations (positions and fitness values); the resolution of fitness value sensors is 1 unit; the ranges of object perception and target perception are rounds.
- Each target: abstracted to be a round with radius of r_t (5 units); robots in the round area will perceive the target; the positions and fitness values are generated randomly and fitness values range from $F_{max} - 2$ to F_{max} (20 units); F_{max} is the range of fitness value sensors or the maximum fitness value of targets.
- Scopes of influence: abstracted to be a series of annuli; the fitness value and width of the innermost annulus is the same as those of the corresponding target; the width of the other annuli is $2r_t$, and the fitness values decrease by 1 unit till 0 from the inside out; the fitness value of overlapped area is the maximum value generated by targets.
- Range of perception: the sensing range of fitness value perception is limited to the current position of the robot; the radius of object perception range is $4r_t$ and the radius of target perception range is r_t.
- Interaction: the range of local interaction is a round and the radius is $4r_t$.
- Maximum speed limit: set to be $2r_t$ to ensure the variance of fitness values in two iterations is less than 2 units.
- Linear acceleration of collection: it takes one robot 10 iterations to collect a target while 10 robots can collect the target in one iteration.
- One problem that has not been considered here is avoiding collisions of robots resulting from route intersection.

III. Environmental Restrictions

In this section, three kinds of environmental restrictions are introduced, which are obstacles, interferences and decoys.

A. Assumptions

- Obstacles: can be perceived, can't be removed from the environment, may obey some distribution, will damage robots involved in collisions; the range of obstacle perception is less than that of object perception.
- Interference sources: can't be perceived, can't be removed from the environment, functionally can be regarded as targets with negative fitness value; the influence of overlapped areas is not less than that generated by each interference source at that area; the fitness value is 0 in areas where interference influence is not less than fitness values.
- Decoys: can be perceived, can't be removed, attract robots and consuming their time or energy.

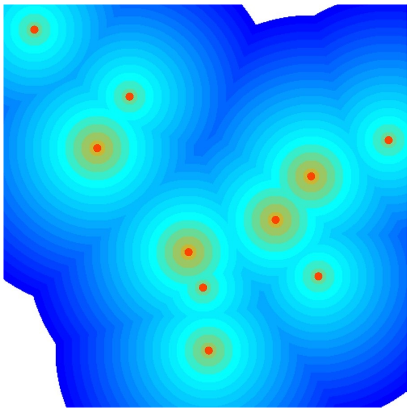


Fig. 1: A screenshot of the problem at the beginning of a simulation. Red rounds stand for the targets. The background color illustrates the fitness of that position. Robots are not illustrated in this figure.

B. An Idealized Model

An idealized model of three environmental restrictions is shown in Figure 2, which is described as following:

- Each obstacle: abstracted to be a square (side length is 1 unit); positions are random; the range of obstacle perception of robots is a round, and the radius is $2r_t$.
- Each interference source: abstracted to be a round with radius of $2r_t$; the positions and interference values are generated randomly, and interference values range from $-F_l$ to $-2F_l$; F_l (5 units) is the maximum influence value of targets; scopes of interference are abstracted to be a series of annuli, and the width of each annulus is $2r_t$; the influence of interference sources is obvious.
- Each decoy: abstracted to be a round with radius of r_t, can be regarded as a target which can't be collect, will not damage robots; the positions and fitness values are generated randomly, and fitness values range from $F_{max} - 3$ to $F_{max} - 1$.

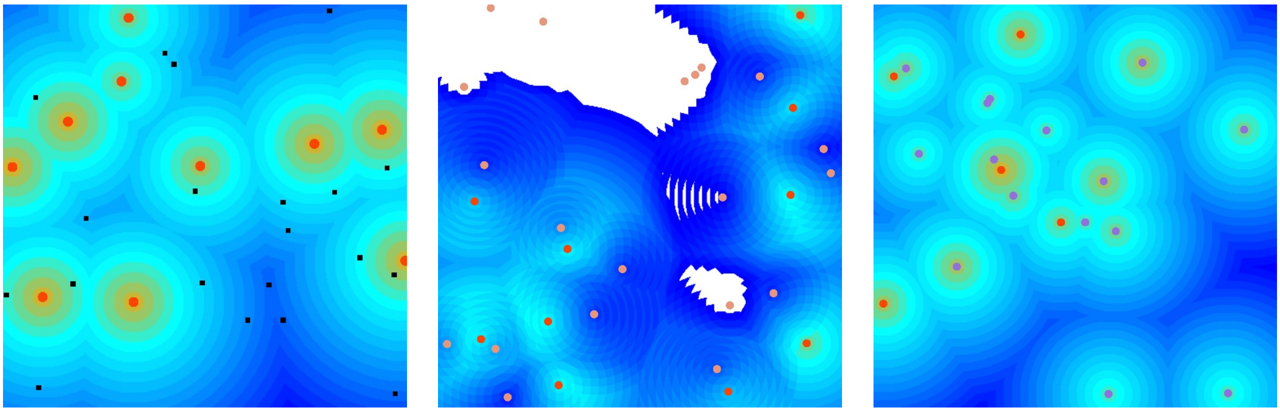
IV. Strategies for Searching

In this section, the group explosion strategy (GES) and improved group explosion strategy (IGES) designed for searching multiple targets are explained in detail.

A. Group Explosion Strategy

1) Overview of Group Explosion Strategy: A flow chart of GES is shown in Figure 3. In order to make full use of the intra-group cooperation and inter-group parallelism, a pre-defined threshold β_G is applied to control the group size.

2) Group Search: When the size of group is within the threshold β_G, the strategy is used to move the group center to the best position within the group.



(a) Problem with Obstacles (b) Problem with Interference sources (c) Problem with Decoys

Fig. 2: A screenshot of the searching problem at the beginning of a simulation. Red rounds stand for the targets, orange rounds stand for interference sources and purple rounds strand for decoys. Black squares are obstacles. The background color illustrates the fitness of that position. Robots are not illustrated.

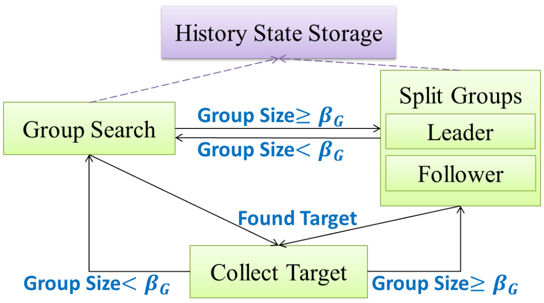
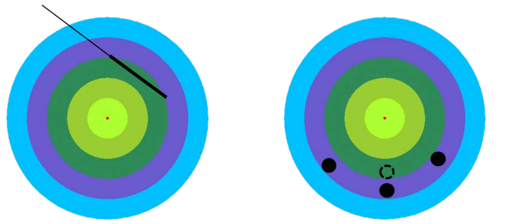
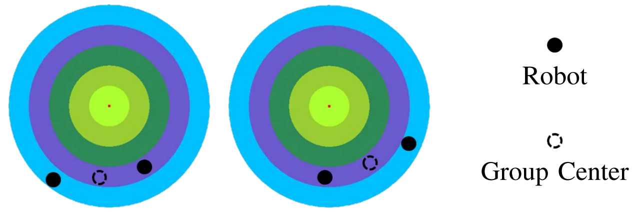


Fig. 3: Flow chart of GES



(a) One robot (b) Robots with same fitness



(c) Infinite Loop between two states

Fig. 4: Three situation which GES does not perform well.

3) Split Groups: When the group size exceeds the threshold β_G, the strategy is splitting the group into two smaller ones. Robots with the best two fitness values, denoted as $L1$ and $L2$, repel each other away while each of the other robots selects a leader to follow independently and randomly.

B. Improved Group Explosion Strategy

The GES shows some drawbacks in certain cases. Thus IGES is proposed, with simpler strategy, fewer parameters and better performance.

1) Shortcomings of Group Explosion Strategy: The basic idea of intra-group cooperation in GES is that the group moves the center towards the best individual of the group, and if multiple robots shares the same fitness value, a random one is picked, which may let robots get stuck or fall back to places with worse fitness value in certain cases, such as the three situations shown in Figure 4.

In Figure 4a, when only one robot is in the group, the robot may bounce along the black line, since all the best positions in history are in this line. In Figure 4b, when several robots share the same fitness value, the group center may be a better position, but the strategy will move the center towards a robot. If the best robot selected is the bottom one, the whole group moves away from the target. In Figure 4c, an infinite loop may occur when the group consists of two robots with different fitness values.

2) Improved Group Explosion Strategy: To solve the problems mentioned above, we proposed four simple but effective strategies which reflect the core ideas of the IGES.

- Strategy 1: This strategy is used for splitting groups when the group size exceeds β_G or members of the group share the same fitness value. The robots in the group are supposed to leave the group center.
- Strategy 2: This strategy is used for robots with different fitness values. The strategy moves the group center towards the center of best positions in the group.
- Strategy 3: This strategy is used when only one robot in the group and the current position is the best in history. And the strategy component equals current velocity.
- Strategy 4: This strategy is applied when there are better fitness values in history. The robot will move towards

the center of best positions in history.

The velocity update equation for robot i is shown below:

$$V_i(t) = S_i(t) + R_C * R_p \tag{1}$$

where $S_i(t)$ is the velocity update vector from the strategy adopted, R_C is a scaling factor shown in Table I and R_p is a unit random vector.

A brief summery of the IGES is shown in Table I

TABLE I: Brief Summary of the IGES

Group Size	Fitness Condition	Strategy	R_C
$\geq \beta_G$	Different Fitness	No. 1+2	$^1/_{10}$
$\in [2, \beta_G)$	Different Fitness	No. 2	$^1/_{10}$
≥ 2	Same Fitness	No. 1	$^1/_{10}$
	Best in history	No. 3	0
$= 1$	Worse than last time	No. 4	1
	Better history in the earlier	No. 4	$^1/_{10}$

It's necessary to note that "No.1+2" means the velocity update vector is the sum of components of strategy 1 and strategy 2. When the group size is 1, the last two situations have the same strategy but different R_C so as to avoid the shortcoming shown in Figure 4c.

V. Strategies for Restriction-Handling

Three environmental restrictions are described precisely in section III, and this section focuses on tackling two restrictions: obstacles and decoys.

A. A Strategy for Obstacle-avoidance

In this paper, a simple avoiding scheme is applied for GES, IGES and RPSO. The robot will check if it will run into obstacles with the updated velocity $V_t(i)$. If so, a small repulsive force perpendicular to $V_t(i)$ from the obstacle will be added to avoid the collision.

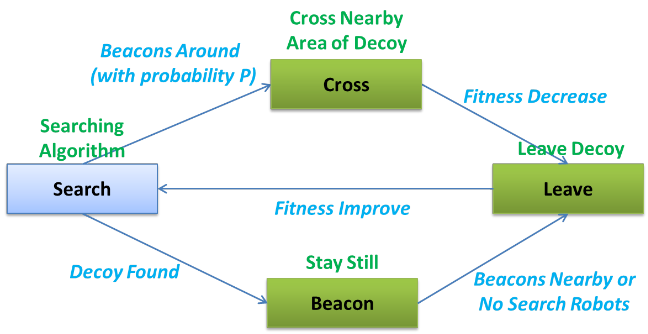
B. Two Strategies for Decoy-Handling

In this section, two strategies are proposed, and the flow charts are shown in Figure 5.

1) Cooperative Strategy: In this strategy, robots may stay in one of the four states: search, beacon, cross and leave. Generally, robots start running the searching algorithms in the search state. Once a robot finds a decoy, it will go into the beacon state. The beacon robot will stay still and give out signals so that its neighbors can perceive the decoy. The beacon robot will go into the leave state if all its neighbors have got the signal or one of its neighbors has already been a beacon.

- Leave States and Cross States: When a robot is in the leave state, the fitness values perceived are decreasing. Robots can go into the leave state from beacon states or cross states. If robots come from the beacon state, they will go into the leave state just by leaving the nearby decoys in a random direction selected uniformly from angle 0 to 2π. When a robot in the search state senses the beacon signal, it will go into the cross state with a possibility P. And in the cross state, the robot will try to cross the area around the decoy with the hope of finding targets nearby.
- Direction Selection in Cross States: In the cross state, the maximum distance between the robot and the decoy is the radius of the interaction range, and the angle from the selected direction to the line between the robot and the decoy should be within the range of $[\pi/6, \pi/3] \bigcup [-\pi/3, -\pi/6]$. The reason for choosing $\pi/3$ and $\pi/6$ as the borders is that directions with angles inside the range have the best efficiency for searching the target in this area.

2) Non-Cooperative Strategy: In this strategy, if a robot finds a decoy, it will go into the leave state and select a random direction to move until the fitness value increases. This strategy is used as the baseline for decoy avoiding.



(a) Flow chart of the Cooperative Strategy

(b) Flow chart of the Non-Cooperative Strategy

Fig. 5: Flow charts of the two strategies. Green texts indicate the behaviors of robots in current states, blue italic texts and arrows indicate the conditions for state transitions.

VI. Simulation Results and Discussion

In this section, three kinds of experiments were carried out, which are used for analyzing parameters, evaluating the performances of searching strategies and testing the effects of restriction-handling strategies respectively. In each test, 20 random maps are generated and each method is repeated for 20 times. And the results in this section are the average results of these 400 runs.

A. Underlying Algorithms

Three underlying algorithms are used in this paper: GES, IGES and RPSO (Robotic Particle Swarm Optimization). GES and IGES are described in section IV. In RPSO, each robot acts as a particle of the PSO and the topology of robots for calculating gbest is spacial-based, and in case of robots' vibrating in an area, a small random unit vector R_p is introduced if both pbest and gbest are the current position.

B. Parameter Analysis

There are two parameters needed to be analyzed: one is the threshold β_G for controlling the group size in GES and IGES, and the other one is in the cooperation strategy of decoy avoidance, the possibility P with which robots in search states go into cross states when they sense the beacon signals.

1) The Threshold β_G of Group Size in GES and IGES: In GES, there are three parameters to be optimized: β_G, β_S and β_R, and they are 6, 0.27, 0.88 respectively when the algorithm is the most efficient in our experiments. In IGES, the threshold β_G is the only parameter needed to be tuned, which plays the same role as that in GES. The number of robots and targets are all 30, and the β_G is within the range of $[4, 30]$. The evaluation criterion is the iterations when the swarm find and collect all the targets, as is shown in Figure 6.

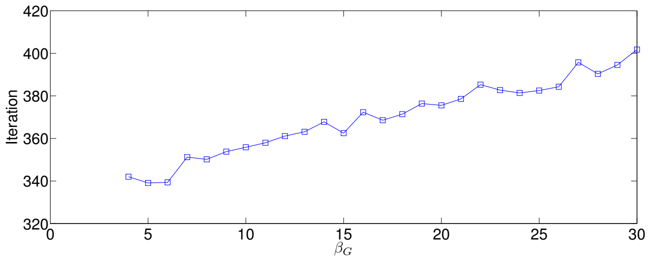


Fig. 6: Parameter analysis result of β_G in IGES.

As is shown, the optimal range of β_G in IGES is similar to that in GES (around 6).

2) The Possibility P of Turning into Cross States: A larger P makes the robot turn into cross states more easily and the robot will spend more time searching for targets nearby the decoy. However, a robot in cross states may miss the targets which are close enough to the decoy, and this situation will be eased if a smaller P is adopted. The parameter is tuned in a fixed environment with 50 robots and 20 targets mixed with 80 decoys. The value of P is selected from the range $[0, 1]$ with a step size of 0.1. According to our experiments, the iterations are not sensitive to P and the optimal value is around 0.8 while the times of decoy visits are influenced greatly by P. A larger P is surely effective on decoy avoidance.

C. Experiments on Searching Strategies

One criterion for measuring the efficiency of searching algorithms, is the iterations needed to collect a certain proportion of targets. Another criterion is the total moving distance of all robots in simulation, which represents part of energy consumption. The last criterion is the average CPU time for updating velocities of all robots per iteration, which is also an important cause of energy consumption.

In our experiments, the numbers of targets and robots are both 10, 30 and 50 respectively, and the proportions of targets adopted are 50% and 100%. According to our experimental results, the IGES dominates in all three criteria. As to the iterations, the GES performs better than the RPSO, which indicates the effectiveness of explosion scheme. And the total distance of RPSO is a little shorter than that of GES. The IGES has the shortest CPU time among three algorithms, $10-30\%$ and $25-50\%$ quicker than RPSO and GES respectively.

To study the performance of three algorithms in more detail, we carried out a series of scalability experiments. The IGES shows great advantage in iterations, and it only need $50-60\%$ and $45-70\%$ of iterations needed by the RPSO and GES. The GES performs better than RPSO in most cases.

D. Experiments on Problem with Environmental Restrictions

In order to study the effect of restriction-handling s-trategies and the anti-jamming performance of searching algorithms, we did a series of experiments.

1) Obstacle Avoidance: The range of obstacle perception of robots is a round, and the radius is $2r_t$, and robots involved in collisions will be removed from the environment. The number of targets and the population are both 30, and the number of obstacles varies from 0 to 500 with step size of 50. The results of three searching algorithms with the strategy for avoiding obstacles are shown in Figure 7. The iterations can be used to analyze the influence of obstacles on algorithms. And another criterion, the number of remaining robots, is introduced to measure the performance of obstacle avoidance, which is better when larger.

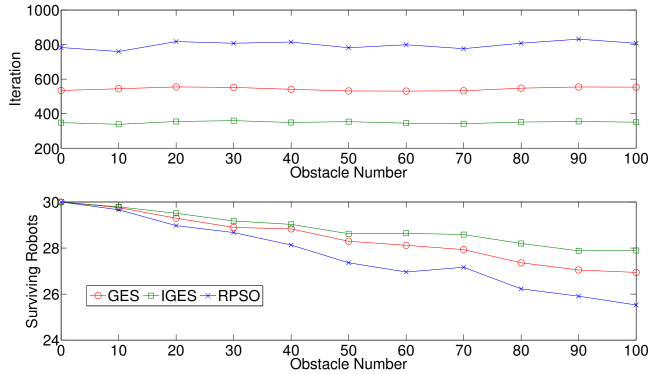


Fig. 7: Results in obstructive environment.

The IGES are better than the other two algorithms on both measures. In the sub-figure of iterations, the waves of the RPSO curve are much bigger than that of GES curve or IGES curve, which means the GES and IGES are more suitable for environment with obstacles. As to the number of remaining robots, the performance of GES and IGES is also better than RPSO.

2) Anti-jamming Performance: The robots can't perceive the interference sources, so it's hard to propose strategies for tackling them. According to our experiments, the IGES shows greater adaptability to interference than comparison algorithms. With the increase in the number of robots and targets, the advantage of IGES is getting more obvious, which indicates the cooperation strategy of IGES performs well in different scales of problems. The interference brings little influence on the performance of IGES (less than 10%).

However, the interference gives a 30% boost in iterations of GES while the change in RPSO is over 50%. The interference sources introduced make the environment more complicated, and the results visually present the adaptability of algorithms to interference, i.e. the GES and IGES are better than RPSO obviously.

3) Decoy Avoidance: In the model stated above, the decoy will not harm the robots, but it may increase the time cost. The experiments on two strategies combined with three algorithms were carried out in the same environment as basic setup except that various numbers of decoys are selected from 0 to 200.

According to our experimental results, the iterations of cooperative strategy is about $5\% - 10\%$ better than that of non-cooperative strategy in three algorithms. Meanwhile, the cooperative strategy shows great performance on the criterion of decoy visit, and the advantage can reach $15 - 25\%$ for GES and $20-30\%$ for RPSO. The cooperative strategy shows better performance yet share almost the same CPU time with the non-cooperative strategy.

In order to study the adaptiveness of the strategies, a series of scalability experiments were carried out. And the overall advantages of cooperative strategy in iteration and decoy visit are quite similar to that in previous experiments. The advantage in decoy visit is much more obvious than that of iteration. When there is more targets or less decoys, the advantage is not obvious, as the definition of cooperative strategy shows that it requires lots of decoys to take effect.

VII. Conclusions

In this paper, we try to give a comprehensive and detained introduction of the multi-target search problem in swarm robotics. In order to promote the study of this problem, we built some idealized models to describe the problem and environmental restrictions more precisely, and brought forward some criteria to evaluate the performance.

In accordance with the basic multi-target search problem, we compared three different searching strategies (GES, IGES and RPSO) in different scales and analyzed their performance. The results show that the IGES is more effective and adaptive than GES and RPSO, which also show the advantage of the explosion scheme.

As to the environmental restrictions, through various scales of experiments with different restrictions, we investigated the performance of three restriction-handling strategies, the anti-jamming performance and extensibility (compatibility with other strategies) of three searching algorithms. And on all measures, the IGES performs better than GES and RPSO, because it makes full use of the intra-group cooperation and the independence of individuals.

As for future work, we plan to improve the searching strategy with more complex yet effective schemes. We also plan to introduce more types of objects that make the model more realistic, such as decoys harmful for robots. And it's also a great challenge to build a mathematical model for the multi-target search problem, which may lead to more rigorous and beautiful theories.

VIII. Acknowledgments

The authors gratefully acknowledge the contributions of ZhongYang Zheng, JunZhi Li and the reviewers.

References

[1] Q. Tang and P. Eberhard, Cooperative motion of swarm mobile robots based on particle swarm optimization and multibody system dynamics, *Mechanics based design of structures and machines*, vol. 39, no. 2, 2011, pp. 179C193.

[2] J.-H. Lee, C. W. Ahn, and J. An, A honey bee swarm-inspired cooperation algorithm for foraging swarm robots: An empirical analysis, *in Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on.*, IEEE, 2013, pp. 489C493.

[3] M. Masár, A biologically inspired swarm robot coordination algorithm for exploration and surveillance, *in Intelligent Engineering Systems (INES), 2013 IEEE 17th International Conference on.*, IEEE, 2013, pp. 271C275.

[4] A. Khapalov, Source localization and sensor placement in environmental monitoring, *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 3, 2010, pp. 445C458.

[5] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, Distributed search and rescue with robot and sensor teams, *in Field and Service Robotics*, Springer, 2006, pp. 529C538.

[6] K. Derr and M. Manic, Multi-robot, multi-target particle swarm optimization search in noisy wireless environments, *in Human System Interactions, 2009. HSI09. 2nd Conference on.*, IEEE, 2009, pp. 81C86.

[7] A. Marjovi, J. Nunes, P. Sousa, R. Faria, and L. Marques, An olfactory-based robot swarm navigation method, *in Robotics and Automation (ICRA), 2010 IEEE International Conference on.*, IEEE, 2010, pp. 4958C4963.

[8] G. Cabrita and L. Marques, Divergence-based odor source declaration, *in Control Conference (ASCC), 2013 9th Asian.*, IEEE, 2013, pp. 1C6.

[9] H. Espitia and J. Sofrony, Path planning of mobile robots using potential fields and swarms of brownian particles, *in Evolutionary Computation (CEC), 2011 IEEE Congress on.*, IEEE, 2011, pp. 123C129.

[10] T. Schmickl and K. Crailsheim, Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm, *Autonomous Robots*, vol. 25, no. 1, 2008, pp. 171C188.

[11] J.-G. Li, Q.-H. Meng, Y. Wang, and M. Zeng, Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm, *Autonomous Robots*, vol. 30, no. 3, 2011, pp. 281C292.

[12] A. Amory, B. Meyer, C. Osterloh, T. Tosik, and E. Maehle, Towards fault-tolerant and energy-efficient swarms of underwater robots, *in Parallel and Distributed Processing Symposium Workshops and PhD-Forum (IPDPSW), 2013 IEEE 27th International*, IEEE, 2013, pp. 1550C1553.

[13] D. Gong, C. Qi, Y. Zhang, and M. Li, Modified particle swarm optimization for odor source localization of multi-robot, *in Evolutionary Computation (CEC), 2011 IEEE Congress on.*, IEEE, 2011, pp. 130C136.

[14] W. Li and W. Shen, Swarm behavior control of mobile multi-robots with wireless sensor networks, *Journal of Network and Computer Applications*, vol. 34, no. 4, 2011, pp. 1398C1407.

[15] Y. Tan and Y. Zhu, Fireworks algorithm for optimization, *Advances in Swarm Intelligence*, 2010, pp. 355C364.

[16] M. Couceiro, R. Rocha, and N. Ferreira, A novel multi-robot exploration approach based on particle swarm optimization algorithms, *in Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, nov. 2011, pp. 327C332.

[17] Z. Zheng and Y. Tan, Group explosion strategy for searching multiple targets using swarm robotic, *in IEEE Congress on Evolutionary Computation*, 2013, pp. 821C828.

[18] Z.Y. Zheng, J.Z. Li, J. Li, and Y. Tan, "Improved Group Explosion Strategy for Searching Multiple Targets using Swarm Robotics", *in 2014 IEEE International Conference on Systems, Man, and Cybernetics*, October 5-8, 2014, Paradise Point Resort and Spa, San Diego, California, USA. pp. 1-8.

[19] Z.Y. Zheng, J.Z. Li, J. Li, and Y. Tan, "Avoiding Decoys in Multiple Targets Searching Problems using Swarm Robotics", *in Evolutionary Computation (CEC), 2014 IEEE Congress on*, pp. 784-791.