A Comparative Study of Biology-Inspired Algorithms Applied to Swarm Robots Target Searching

Qirong Tang[1], Lei Zhang[2], Wei Luo[3], Lu Ding[1], Fangchao Yu[1], and Jian Zhang[4](✉)

[1] Laboratroy of Robotics and Multibody System, School of Mechanical Engineering,
Tongji University, No.4800, Cao An Rd., Shanghai 201804, China
qirong.tang@outlook.com

[2] Department of Mechanical Engineering of Tsinghua University,
Graduate School at Shenzhen, Shenzhen 518055, China

[3] Department of Mechanical Engineering, University of Stuttgart,
Pfaffenwaldring 9, 70569 Stuttgart, Germany

[4] Institute of Advanced Manufacturing, School of Mechanical Engineering,
Tongji University, No.4800, Cao An Rd., Shanghai 201804, China
jianzh@tongji.edu.cn

Abstract. In this study, the mechanisms of some creatures' behaviors collaborated in swarm are applied to the coordination of swarm robots, especially for them to search target. Three typical biology-inspired algorithms, i.e., Particle Swarm Optimization, Ant Colony Optimization and Genetic Algorithms, are thus compared, systematically. Corresponding tasks and mathematical models are set up. Based on the experimental work within MATLAB, the performances of the concerned algorithms on the difficulty of task mapping, adaptability for various terrains, as well as convergence and stability are elaborately analyzed and verified, which is helpful for designing real physical swarm robotic systems.

Keywords: Swarm intelligence · Swarm robots · Obstacle avoidance · Particle swarm optimization · Ant colony · Genetic algorithms

1 Introduction

A single robot has restrictions and limitations in acquiring and processing information, thus it is not suitable to deal with complicated works and face to the fast changing environments. However, this situation motivated many researchers to start the study of multi-robot system at the 1980s. Gradually, two specialized subjects, namely swarm robotics and swarm intelligence, were born. Under collaboration, a swarm of robots with inferior abilities for each can fulfill some complicated tasks that a single but superior and expensive robot cannot. As for now, many robots such as e-Puck [1], s-bot [2] are designed and created for the

Y. Tan et al. (Eds.): ICSI 2016, Part II, LNCS 9713, pp. 479–490, 2016.
DOI: 10.1007/978-3-319-41009-8_52

research of swarm robots. A swarm of simple robots can be designed and manufactured more easily and cost less. Moreover, swarm robots also outperform on robustness, scalability, flexibility [3]. Therefore, swarm robots are used widely.

Swarm robots demonstrate novelties on collaboration among massive robots, which are often inspired from biologically collective behaviors. In term of mechanisms of these collective behaviors, many swarm intelligent algorithms have been constructed, such as, Particle Swarm Optimization/PSO [4], Ant Colony Optimization/ACO [5], Genetic Algorithms/GAs [6], and Bacterial Forging Algorithm [7]. Having neither centralized control nor complete environment information, swarm intelligence technology brings a new way of solving complicated problems by taking the advantage of its scalability. Up to now, swarm robotics research has accumulated some benchmark tasks such as aggregation, area coverage, mapping, searching, navigating and locating leakages [8–10].

Different swarm intelligent algorithms have different features. However, to the best of our knowledge, the research of a systematically comparison of these algorithms applied to swarm robots have rarely been studied. So this study aims to discover the differences of PSO, ACO and GAs when they are applied to swarm robots in searching one or more targets.

Section 2 presents the task mapping models and mathematical models of the studied algorithms as well as some improvements. In Sect. 3, it makes a quantity of simulation in various environments containing obstacles and analyzes the performance of the three concerned algorithms. Conclusions are given in Sect. 4.

2 Models

2.1 Particle Swarm Optimization

The task environment of this work is a two-dimensional space with obstacles. The velocity of the particles in PSO are updated by

$$V_i^{k+1} = \omega V_i^k + c_1 r_{i,1}^k (X_{i,self}^{best,k} - X_i^k) + c_2 r_{i,2}^k (X_{i,local}^{best,k} - X_i^k) \ , \tag{1}$$

where X_i^k is the position of particle i at kth iteration, V_i^k is the velocity of particle i at kth iteration. One saves $X_{i,self}^{best,k}$ as the individual best position, and $X_{i,local}^{best,k}$ is the best position among particles. Here ω is an inertia weight, c_1, c_2 are referred to as cognitive scaling and social scaling factors, $r_{i,1}^k$, $r_{i,2}^k$ are two independent random parameters. The position of particle i is given by

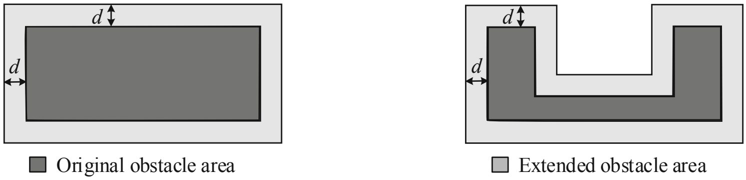
$$X_i^{k+1} = X_i^k + V_i^{k+1} \Delta t \ , \tag{2}$$

where Δt is a time step that usually is 1 and thus is omitted often.

Parameter ω represents the inertia of the particles, and be calculated by

$$\omega = \frac{(\omega_2 - \omega_1)}{e} \times j + \omega_1 \ . \tag{3}$$

Here e is the maximum steps, j is the current step, $\omega_1 = 0.4$, $\omega_2 = 0.9$.



(a) A convex obstacle (b) A non-convex obstacle

Fig. 1. Type of obstacles

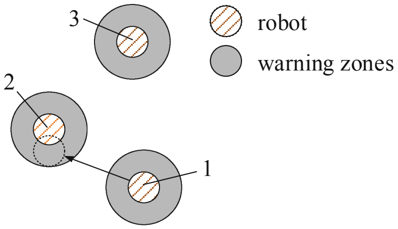


Fig. 2. Choice of rotation direction

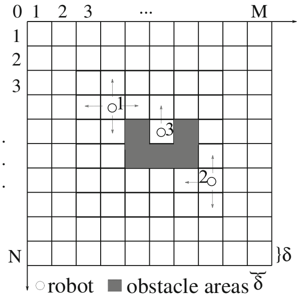


Fig. 3. The searched environment

The shape of obstacles in real environments is irregular. We shape obstacles with regular boundaries, and categorize them into convex and non-convex, as shown in Fig. 1. Considering braking and signals delay, the extended obstacle areas are set up [11]. When one robot detects obstacles in front, it responds to avoid obstacles by slowing down and rotating itself. This study takes 10° for each rotation.

When updating positions, these robots with physical volume should pay attention so as to keep a safe distance to their peers. In Fig. 2, a mutual collision between robot 1 and robot 2 may happen since robot 1 is entering the warning zone of robot 2. Similar to static obstacles in practical environment, robot 1 takes actions to avoid collision with robot 2. After several rational tries (maximum 18 times), if the robot still can't find a suitable collision free path, it will stay there instead of updating for the current step. In this study, one robot is represented by one particle.

2.2 Ant Colony Optimization

Here, the robot is assumed as a circular with radius R and searches in a two-dimensional space which is divided into an $M \times N (M = N = 10)$ grid, see Fig. 3. All grids have the unit length δ, and consist of barrier grids and free grids. Barrier grids can be used to complement the irregular boundary of the environment area as well, besides of indicating obstacles. One obstacle occupies one or more grids. If some part of the obstacle is not enough to fill a grid, it will be considered as a complete grid. According to [12], if a robot stands at node i, its next node j is governed by

$$j = \begin{cases} argmax\{\tau_{i,s}^{\alpha}(t)\eta_{i,s}^{\beta}(t)\}, s \in A, q \leq q_0 \\ P_{i,s} = \dfrac{\tau_{i,s}^{\alpha}(t)\eta_{i,s}^{\beta}(t)}{\sum\limits_{s \in A} \tau_{i,s}^{\alpha}(t)\eta_{i,s}^{\beta}(t)}, else \end{cases} , \tag{4}$$

where $q \sim (0, 1)$ is an random variable, q_0 is a constant ranges from 0 to 1, s is one element of A which contains all candidate nodes for node i, $\tau_{i,s}^{\alpha}(t)$, $\eta_{i,s}^{\beta}(t)$ represent the pheromone concentration and the heuristic information, respectively, from node i to node s in which $\tau_{i,s}^{\alpha}(t)$ and $\eta_{i,s}^{\beta}(t)$ are corresponding coefficients. Here $argmax\{\tau_{i,s}^{\alpha}(t)\eta_{i,s}^{\beta}(t)\}$ means the corresponding node s reach the maximum of $\tau_{i,s}^{\alpha}(t)\eta_{i,s}^{\beta}(t)$, $P_{i,s}$ is the probability that node s is selected as the next position for node i. When the robot is trapped into a valley (see node 3 in Fig. 3), it sets the node 3 as a barrier grid, and goes back to the previous node.

The heuristic information is calculated according to a heuristic algorithm summarized by

$$\eta_{i,s} = D/d_{s,G} \ , \tag{5}$$

where D is the weight coefficient and $d_{s,G}$ indicates the distance between the node s and target node G. After one iteration, the pheromone concentration needs to be updated by

$$\tau_{i,j}(t+1) = (1-\varphi)\tau_{i,j}(t) + \varDelta\tau_{i,j} \ , \tag{6}$$

$$\varDelta\tau_{i,j} = \sum_{k=1}^{m} \varDelta\tau_{i,j}^{k} \ , \tag{7}$$

where $\tau_{i,j}$ represents the pheromone concentration on the path from node i to node j before updating, φ refers to the pheromone evaporation factor, and m is the robot number, $\varDelta\tau_{i,j}$ is the pheromone that robot k leaves on the path from node i to j, $\varDelta\tau_{i,j}$ is the sum of pheromone of all robots at current iteration which is a constant Q in this study.

Each robot selects the next node from the four directly adjacent grids. Therefore, the path is composed by a series of perpendicular segments. An optimized path will be obtained when the redundant nodes are deleted, see Fig. 4.

2.3 Genetic Algorithms

Genetic algorithms are applied to solve robot path planning in this section via using the grids map model as same as for ACO. One chromosome represents one path and the genetic information on which is the coordinate values of path nodes. All these path nodes form the complete path.

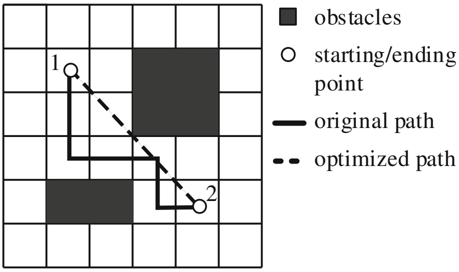


Fig. 4. Remove redundant nodes

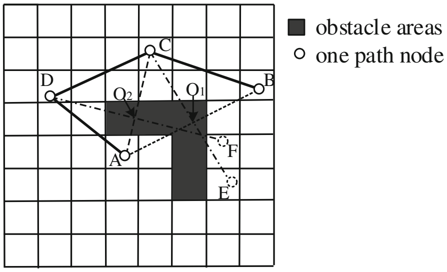


Fig. 5. The geometric method

A geometric obstacle avoiding method is proposed in [13] by inserting some nodes along the direction perpendicular to the connection line of two nodes. The inserted nodes should be on the same side. This study proposes a new method for judging if the later node is on the same side with the formers. In Fig. 5, since the path from node A to node B collides to one obstacle, node C is inserted firstly, which is in the free area and on the line perpendicular to the connection between node A and node B. After this, node D rather than node F, is inserted since the sign of $\overrightarrow{AD} \cdot \overrightarrow{DC}$ is same to that of $\overrightarrow{AC} \cdot \overrightarrow{CB}$, and the sign of $\overrightarrow{AF} \cdot \overrightarrow{FB}$ is in opposite to $\overrightarrow{AC} \cdot \overrightarrow{CB}$. Finally, a free path $A \rightarrow D \rightarrow C \rightarrow B$ is generated.

To prevent the optimal path from destroying in GA operation, the selection operator endowed with elitist strategy is considered. This study uses the crossover of finding an intersection point(node) of two parent paths, then exchange mutually all of the rest path nodes after this node. And one-point mutation is adopted, which randomly choose one node on the original path, then replace it with another node which is near to the original one and locates in a free grid. The insert mode of genetic operator is chosen as that those outstanding individuals are added into parent population to generate a new one.

3 Simulation and Comparison

Parameter settings of the three algorithms are listed in Table 1. The simulation environment is set as 10m×10m. All of these parameters for each algorithm are carefully selected to fit the current condition and make each algorithm currently optimized.

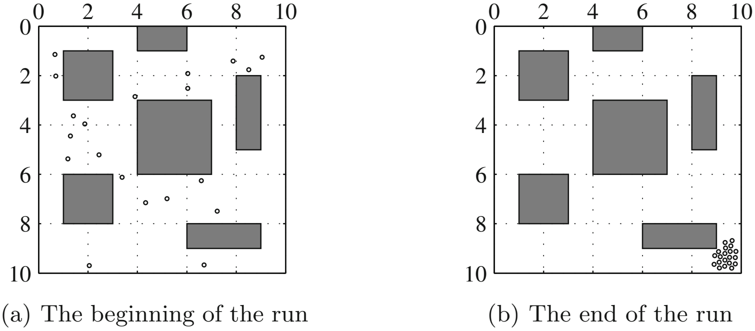
3.1 Comparison of Environmental Adaptability

The terrains are usually very complex in practical environment, thus the environment adaptability of robot has significant influence on the accomplishment of tasks. This study sets up three kinds of environments to test the concerned algorithms.

(1) Comparison in the environment with convex obstacles

Table 1. Parameters list for experiments

Types	Parameters	Values
PSO	maximal velocity (m/s)	0.1
	safety distance between robots (m)	0.3
	safety distance between robot and obstacle (m)	0.2
	inertial parameter (ω)	$0.4 - 0.9$
	individual weight (c_1)	1.0
	neighbor weight (c_2)	1.0
ACO	pheromone increment factor	0.001
	pheromone evaporation factor	0.1
	probability selection parameter (q_0)	0.8
	pheromone concentration weight (α)	1
	heuristic information weight (β)	1
GAs	selection probability	0.9
	cross probability	0.7
	mutation probability	0.3

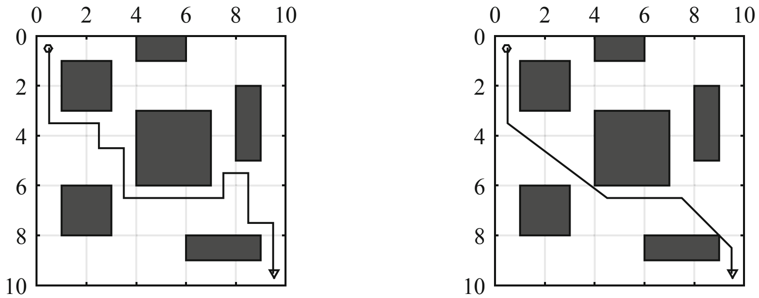


(a) The beginning of the run

(b) The end of the run

Fig. 6. The motion process of all robots

The first environment contains only convex obstacles, see Figs. 6, 7, 8, and 9. The target is purposely placed at location $(9.5, 9.5)$. The initial position and velocity of PSO are randomly generated, while for GA and ACO the initial points are set at the location $(0.5, 0.5)$, which are far away to the target. Figure 6 shows a running progress with 20 particles based on PSO. At the very beginning, the particles are randomly distributed. With the progress of iteration, all particles gather near to the target and keep a certain distance from each. Figure 8 shows the trajectory of 3 particles (robots), and obviously indicates that all robots have successfully avoided the static obstacles based on PSO. Figure 7 shows the result based on ACO, in which "◯" and "▽" represent the start point and the target, respectively. It is obvious that the path in Fig. 7(b) is more smooth than that in Fig. 7(a). GA has also gained a collision free path, see Fig. 9.



(a) The original path

(b) The optimized path

Fig. 7. Path planning result based on ACO

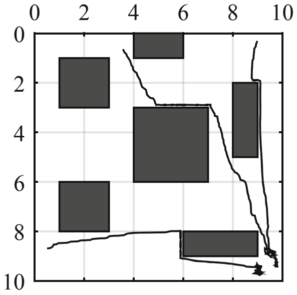


Fig. 8. Trajectories of robots based on PSO

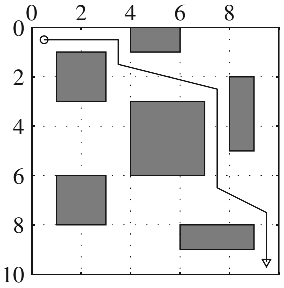


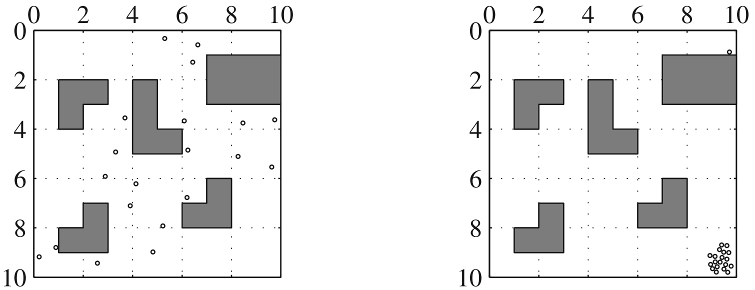
Fig. 9. Path planning based on GA

(2) Comparison in mixed-obstacle environment

In this kind of environment, there are convex and non-convex obstacles which increase the difficulty of target searching. In Fig. 10, most of particles based on PSO have reached the target after certain iterations, but a few of particles fail and fall into corners, like the top-right corner in Fig. 10(b). It indicates that PSO has limitation to handle this situation. It is even more restricted in an environment with U-shaped obstacles. Ant colony optimization also performs well in the second environment, see Fig. 11. Compared with the path with redundancy nodes, the optimized one is better on the path length and smoothness, which shows the necessity of removing redundant nodes. Figure 12 displays the path planning result of GA in mixed-obstacle environment. It indicates that the geometric method for obstacle avoidance has a strong generality and adaptability to the environment with convex and non-convex obstacles.

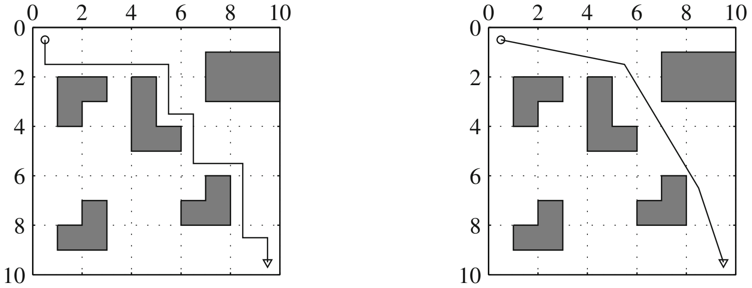
(3) Comparison in the environment with U-shaped obstacles

Compared to the previous two types of obstacles, the U-shaped obstacle brings a lot of difficulties for target searching. To verify the search ability of algorithms as possible as one can, in this paper the initial point is set inside the U-shaped area, at (4.5, 4.5), and the target point is set at outside of the



(a) The start of the run (b) The end of the run

Fig. 10. The motion process of all robots based on PSO



(a) The original path (b) The optimized path

Fig. 11. Path planning result based on ACO

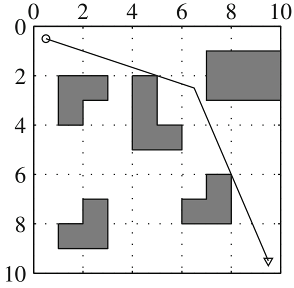


Fig. 12. Path planning of GA in mixed-obstacle environment

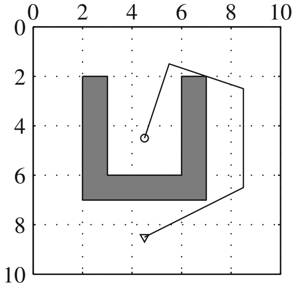
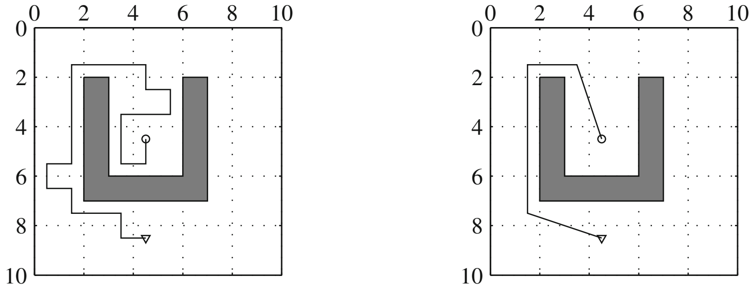


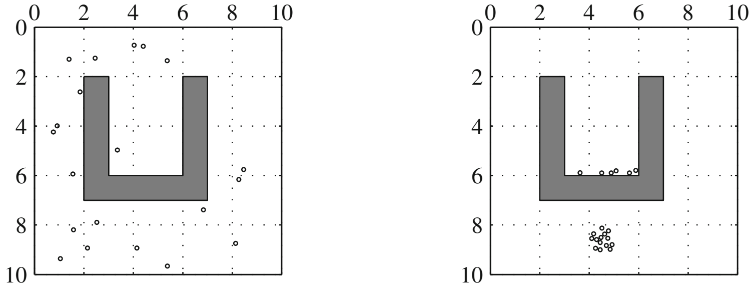
Fig. 13. Path planning of GA in U-shaped obstacle environment

U-shaped, at (4.5, 8.5). Other conditions are same to the previous cases. As shown in Figs. 13 and 14, GA and ACO are also feasible to the big U-shaped obstacle. In Fig. 15, most of particles have arrived the target point, but there are still a certain particles stuck in the U-shaped obstacle.



(a) The original path (b) The optimized path

Fig. 14. Path planning result based on ACO



(a) The start of the run (b) The end of the run

Fig. 15. The motion process of all robots based on PSO

3.2 Convergence Comparison of the Studied Three Algorithms

The convergence speed is one of the evaluation criteria of algorithms and affects to the computational efficiency. As the iteration runs, the trend of the optimal solution can reflect the convergence feature. In Fig. 16, the optimal objective function value approach to 0 after 100 iterations, which means that the target is found. Overall, the convergence trend of PSO is distinct and the converge speed is fast. However, in some complex environments such as the U-shape obstacle environment, some individuals are easily fall into local optimum. ACO has a feature of convergence gradually, shown in Fig. 17. In the first 20 iterations, the path length varies greatly, fluctuates wildly and its convergence trend is not obvious. After all, ACO still converges to the optimal solution at a relatively fast rate. Genetic algorithm hardly converge to an optimum solution, see in Fig. 18, the shortest path is irregularly changing without a tendency of convergence. The reason is that many operations of GA have large randomness, such as the crossover or mutation.

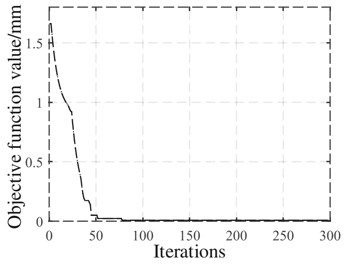


Fig. 16. The convergence of PSO

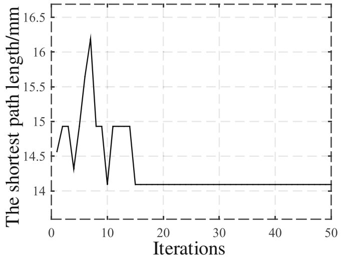


Fig. 17. The convergence of ACO

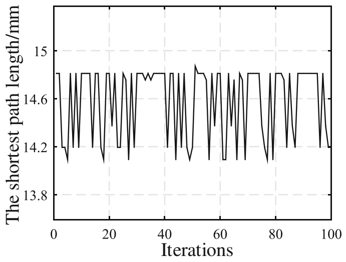


Fig. 18. The convergence of GA

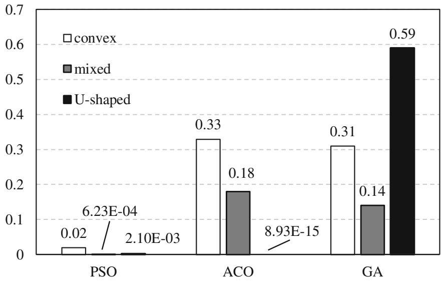


Fig. 19. Stability comparison

3.3 Stability Comparison of the Studied Three Algorithms

Stability is another important evaluation criterion of the algorithms. Each of three algorithms is evaluated independently with 100 runs in the three types of environments. As one can see from Fig. 19, PSO has a good stability in three environments, especially for the mixed-obstacle environment, which has only a standard deviation of 6.23E-04. It shows that swarm robots based on PSO can find stably the target from the perspective of swarm performance. The stability of ACO in the environment with U-shaped obstacles is much better than that in other two environments. The reason is that there are more obstacles in the latter two environments and more various paths are produced after optimization. Therefore, the shortest path is varying heavily. Genetic algorithm demonstrates the instability in three environments on account of its large standard deviation that is related to its big randomness.

4 Conclusion

In this study, PSO, ACO and GAs are applied to the swarm robots target searching and path planning. Through a quantity of simulation and analysis, the performances of the three algorithms are compared on adaptability, convergence and stability.

The results show that the convergence and stability of PSO are good in the environment with convex obstacles. However, when it comes to non-convex obstacles and large U-shaped obstacles, the basic PSO has difficulty in solving this kind of problem in which particles are easily trapped at obstacles. Ant colony algorithm has a good adaptability to the environment change in this study, but it mainly aims at discrete space and its calculation time is longer than others. Compared with the other two, GAs model is more complex and the quality of initial population has a great influence on the final solution. The geometric method for obstacle avoidance which is used in this study has advantages of generality and good adaptability. The future work will concentrate on including more properties of practical robots into the studied algorithms.

Acknowledgements. This research is supported by the Fundamental Research Funds for the Central Universities (No. 2014KJ032, 20153683), by Shanghai Pujiang Program (No. 15PJ1408400) and the Key Basic Research Project of 'Shanghai Science and Technology Innovation Plan' (No. 15JC1403300). Meanwhile, this work is also partially supported by the State Key Laboratory of Robotics and Systems (Harbin Institute of Technology) (No. SKLRS-2015-ZD-03), and the National Science Foundation of China (No. 51579053). All these supports are highly appreciated.

References

1. Cianci, C.M., Raemy, X., Pugh, J., Martinoli, A.: Communication in a swarm of miniature robots: the e-Puck as an educational tool for swarm robotics. In: Şahin, E., Spears, W.M., Winfield, A.F.T. (eds.) SAB 2006 Ws 2007. LNCS, vol. 4433, pp. 103–115. Springer, Heidelberg (2007)
2. Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.W., Floreano, D., Deneubourg, J.L.: Swarm-bot: a new distributed robotic concept. Auton. Robots **17**, 193–221 (2004)
3. Şahin, E.: Swarm robotics: from sources of inspiration to domains of application. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics 2004. LNCS, vol. 3342, pp. 10–20. Springer, Heidelberg (2005)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: International Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
5. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperative agents. Phys. Rev. Lett. **75**, 2686–2689 (1995)
6. Zhou, L.F., Hong, B.R.: A knowledge based genetic algorithm for path planning of a mobile robot. Acta Electronica Sin. **5**, 4350–4355 (2006)
7. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. **22**, 52–67 (2002)
8. Soysal, O., Şahin, E.: A macroscopic model for self-organized aggregation in swarm robotic systems. In: Şahin, E., Spears, W.M., Winfield, A.F.T. (eds.) SAB 2006 Ws 2007. LNCS, vol. 4433, pp. 27–42. Springer, Heidelberg (2007)
9. Payton, D., Estkowski, R., Howard, M.: Pheromone robotics and the logic of virtual pheromones. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics 2004. LNCS, vol. 3342, pp. 45–57. Springer, Heidelberg (2005)

10. Penders, J., Alboul, L., Witkowski, U., Naghsh, A., Saez-Pons, J., Herbrechtsmeier, S., El-Habbal, M.: A robot swarm assisting a human fire-fighter. Adv. Rob. **25**, 93–117 (2011)
11. Tang, Q., Eberhard, P.: A PSO-based algorithm designed for a swarm of mobile robots. Struct. Multi. Optim. **44**, 483–498 (2011)
12. Luo, D., Wu, S.: Ant colony optimization with potential field heuristic for robot path planning (in Chinese). Syst. Eng. Electron. **32**, 1277–1280 (2010)
13. Li, Q., Feng, J., Liu, Y., Zhou, Z., Yin, Y.: Application of adaptive genetic algorithm to optimum path planning of mobile robots (in Chinese). J. Univ. Sci. Technol. Beijing **30**, 316–323 (2008)