# A PSO-Inspired Multi-Robot Search Algorithm Independent of Global Information

Qian Zhu
School of Software and Shanghai Key Laboratory of Scalable Computing and Systems
Shanghai Jiao Tong University
Shanghai, China
zq_best@sjtu.edu.cn

Alei Liang
School of Software and Shanghai Key Laboratory of Scalable Computing and Systems
Shanghai Jiao Tong University
Shanghai, China
liangalei@sjtu.edu.cn

Haibing Guan
Department of Computer Science and Shanghai Key Laboratory of Scalable Computing and Systems
Shanghai Jiao Tong University
Shanghai, China
hbguan@sjtu.edu.cn

**Abstract—This paper presents an algorithm that coordinates mobile robots to find the desired targets without depending on precise global information. We compare the abstract solution space in PSO and physical environment that robots will explore. According to similarities and differences between them, we introduce a PSO-inspired search algorithm to guide robots to complete the search mission. Moreover, a novel method based on Cartesian geometry for unifying relative coordinate systems will be adopted to improve system's robustness and efficiency.**

***Keywords-multi-robot search; PSO-inspired; relative coordinate system; repulsion;***

## I. INTRODUCTION

The amount of research on robotics application for targets searching has grown substantially. Robots equipped with sensors and wireless telecom devices move around an area to search for some given targets. It is more preferable when the area is dangerous or inaccessible to human. For example, [1] apply robots to locate mines. [2] dispatch robots to search and rescue victim in disaster area. [3] explore planet with robots. Comparing with robot system under central node control, the distributed mobile multi-robot system is more attractive due to its robustness, flexibility, and reliability. The breakdown of any robots would not degrade performance of the whole system severely. The scalability of system ensures the additional robots will increase efficiency to complete search mission.

At a high level of abstraction, optimization problem and multi-robot search task have the same goal: Get some points that best fit the requirements. As Particle Swarm Optimization(PSO) which belongs to Swarm Optimization field demonstrated its great power in solving complex and high-dimensional optimization problem , many algorithms inspired by PSO has been adopted to solve multi-robot search problem in the field of Swarm Robotics. They can be separated into three kinds:

*1*). Applying PSO to optimize the cooperative algorithm. In [8], robots are assigned to search for multiple targets and push them to a specific location; PSO is used to adjust the balance between the exploration and exploitation. [14] apply PSO to update the parameters of robotic search algorithm inspired by chemotaxis behavior in bacteria.

*2*). Modeling robot's movement and corporation with PSO. Each robot is considered as a unique particle in swarms. Its pattern of moving, sensing, and communicating is set according to PSO formulation. [15]demonstrates the use of a distributed PSO algorithm with a novel adaptive RSS weighting factor to guide robots for locating targets in high risk environments. [5] deploy some robots to locate odor source in dynamic environment that will change randomly according the wind speed and its direction. The authors modify PSO by incorporating the change detection and responding mechanisms for solving dynamic problems. [4] apply distributed PSO to a robotic swarm search. The robots determine their position by triangulating from three cricket motes set up as beacons.

*3*). Combing two methods above. They model each robot's micro-state with PSO formulation and optimize it's parameters with PSO. [7] apply PSO to model multi-robot search and quantifies the effectivity of each parameters. It assumes that robots can get the intensity of the target signal which is in inverse proportion to the square of the distance between robot and target. [6] coordinate robots by an outer PSO, and use an inner PSO to optimize the parameters of outer PSO. They define each robot's fitness as the strength of signal from target. Personal and global best[12] are calculated based on it.

As above mentioned, most previous work assume that every robot in search mission is able to get two kinds of global information: absolute location information of itself [4] [6] [7] [8] and some sort of stable pervading signal from target [6] [7] [14]. But under the actual environment, these preconditions are always absent. GPS would not do any favor if the search task is carried out indoor. We cannot even get the precise location information outdoor with the help of GPS whose signal is affected by the weather condition. Besides, most of things in our daily lives do not send any signal that robots can receive. For example, if the search target is a book, a key, or water which can be detected unless staying close enough to it, how robots get its personal best and global best as defined before? These limitations from objective and physical environment would not be encountered in ideal particle swarm optimization. In this paper, we will analyze the similarities and differences

This work was partly supported by the National Natural Science Foundation of China under Grant 60773093, 60970107, 60970108 and the Science and Technology Commission of Shanghai Municipality under Grant 09510701600, 10DZ1500200.

between optimization problem and multi-robot search task. According to these similarities and differences, a PSO-inspired distributed cooperative algorithm will be elaborated in detail.

The paper is organized as follows: Section II compare multi-robot search task in actual environment and particle swarm optimization in abstract solution space. Section III presents a method for unifying relative coordinate systems which is the basis of our cooperative algorithm. Section IV describes a PSO-inspired distributed cooperative algorithm independent of accurate global information for multi-robot search task. Section V analyze the simulation result using Player/Stage—a multi-robot simulation system developed by Brian Gerkey, Richard Vaughan, Andrew Howard, and Nathan Koenig [9].Section V summarizes the research conclusion and future work.

## II. COMPARISON

Swarm intelligence became more and more in use during the last 20 years [10]. As an application of this term, Swarm Optimization has been developed significantly. "Particle Swarm Optimization"(PSO) [11] [12] is an outstanding algorithm in this field. But as the original application of Swarm Intelligence, Swarm Robotics grows slowly. One of the reasons is that real environment which is the platform for robots moving, exploring and sensing is more complex and unstable than abstract solution space in the field of Swarm Optimization. [7] presents some differences between particle swarm optimization and multi-robot search task. We elaborate the resemblances and some important difference between these systems as follows.

Resemblance:

- The same goal: Search for some special points in two or more dimensional solution space
- The same computational model: absolutely paralleled, distributed and heuristic rather than sequential, centrally-controlled or deterministic.
- Both of them complete their mission through cooperating and sharing information among a lot of simple nodes.

Difference:

- Robot Collision versus Agreement: Particle Swarm is defined as they might occur in high-dimensional cognitive space, where collision is not a concern. When two particles converge on the same **point** in cognitive space, we call it "agreement" not "collision"[13]. In multi-robot system, both robots and targets occupy particular **positions** in space, so a collision avoidance algorithm is desired to prevent robots from becoming stuck.
- Unified Coordinate Systems on solution space versus Relative Coordinate System in robot: All particles in PSO share the unified coordinate system dependent on the solution space. But in an uncertain environment, the size and shape of area is usually unknown. Each robot is only able to get its position relative to its original point. All this kind of position form robot's relative coordinate system. Unifying different relative coordinate systems in robots is necessary to share information on the area.
- Neighborhood. Particles in PSO is able to share information with other particles that can be anywhere in the solution space. But there are some limitations on the range and capacity of communication among robots. Every robot can only communicate with others within certain fixed range R. So we define R as its neighborhood structure.
- Fitness function: Particles in PSO can get its fitness value computed by the fitness function. Figure 1(a) illustrate fitness landscapes for the one-dimensional function x + 4 = 10, with goodness defined two different ways [13]. In most cases, robots can't get any heuristic cues about target's location unless the distance between them is fairly small. Most common things do not emit any stable pervading signals. So the fitness value of position is always zero, if we calculate it depending on PSO formulation directly. Figure 1(b) is the fitness landscapes for multi-robot search task. We have to find another way to represent personal and local best.

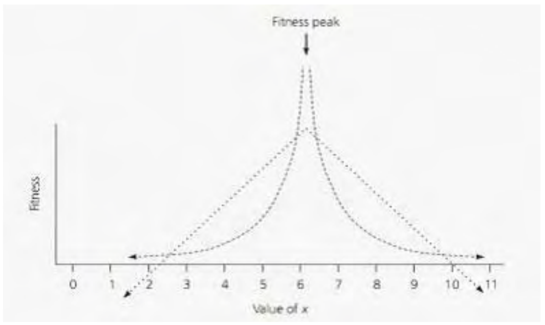

Fig. 1(a) Fitness landscapes for the one-dimensional function x + 4 = 10, with goodness defined two different ways.

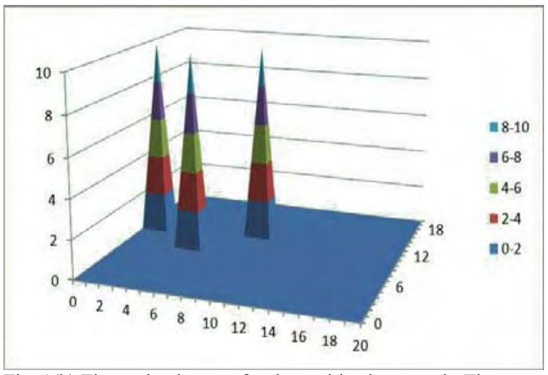

Fig. 1(b) Fitness landscapes for the multi-robot search. The position of targets are (1, 12), (5, 9), (9, 13) respectively. Most of positions' fitness value is zero because robot can't receive any signals form targets.

## III. A METHOD FOR UNIFYING RELATIVE COORDINATION SYSTEMS

In a distributed multi-robot search problem presented here, there are some targets and obstacles randomly distributed within a bounded area. Robots need to find the targets and avoid obstacles through cooperation and sharing information about environment. Both targets and obstacles are static. The search task is completed when all targets has been found.

There are several assumptions for this system. (1)Robot can only get its position relative to the original point while knowing nothing about its global position. Besides, robots do not know the shape and size of search area. (2)When obstacles (including other robots) are very close to robot, they can be detected. (3)Camera on robot can distinguish some characteristic obstacles by their color, shape, etc. This kind of obstacles is taken as reference points. (4)Robot can only communicate with its neighbors within predefined communication range. (5) Targets do not emit any signals that can be received by robots far away. In other words, robot can only find and sense the target when they are close to each other. It is the most common situation.

Communicating with another robot is necessary to obtain local best. Being short of global information, every robot just locally broadcast its message about the area based on its relative coordinate system periodically. The message contains the area has been searched, the position and value of local best, the reference points have been recognized, the number of robots which share the same coordinate system with itself. Unless different relative coordinate systems have merged into a unified one, robot cannot understand the meaning of message from another one. For example, robot A get a message from robot B saying position of local best is (i, j). Robot A does not know where it is. (i, j) is the position in robot B's relative coordinate system. We define that Y axis of relative coordinate system points in the direction in which the robot face initially, and X axis points to the right side (as shown in Fig 2(a)). The X and Y axis with scales form the absolute coordinate system in the simulated environment. Green and yellow spots represent characteristic obstacles and targets respectively. Fig2 (b) is the coordinate system for robot A. The green spots M,N represent obstacles that has been recognized by robot A and B. Fig2(c) is the coordinate system for robot B.As shown in figure 2(b)(c),M and N got different values in different coordinate system. In Fig2 (b), M is (2, 2). N is (3,4). In Fig2(c), M is (0, 0.82), N is (-0.7066, 2.9498).As we will see, M and N serve as the reference objects in unifying coordinate system.

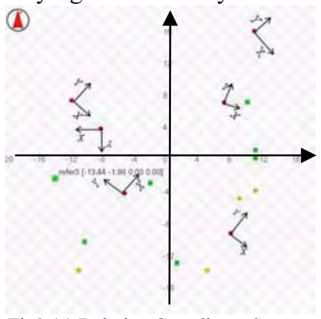

Fig2 (a) Relative Coordinate System Versa Absolute Coordinate System

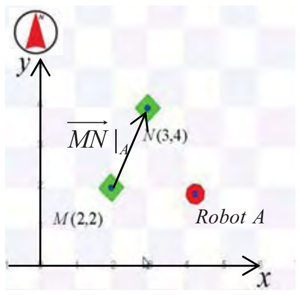

Fig2(b) Coordinate System for Robot A

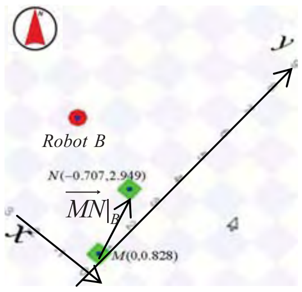

Fig2(c) Coordinate System for Robot B

In coordinated system A( Fig2b ):

$$\overrightarrow{MN}\Big|_A = (N_{xa}-M_{xa}, N_{ya}-M_{ya}) = (\gamma_A,\phi_A)$$

$(\gamma_A,\phi_A)$ is a form of $\overrightarrow{MN}\Big|_A$ corresponds with the polar geometry

$(M_{xa}, M_{ya})$ $(N_{xa}, N_{ya})$ are coordinated value of M and N in coordinated system A respectively.

$\phi_A$ is the degree from $\overrightarrow{MN}\Big|_A$ to X axis.

$N_A$ is the number of robots that have communicated and share the same coordinated system with Robot A.

In coordinated system B( Fig2c ):

$$\overrightarrow{MN}\Big|_B = (N_{xb}-M_{xb}, N_{yb}-M_{yb}) = (\gamma_B,\phi_B)$$

$(\gamma_B,\phi_B)$ is a form of $\overrightarrow{MN}\Big|_B$ corresponds with the polar geometry

$(M_{xb}, M_{yb})$ $(N_{xb}, N_{yb})$ are coordinated value of M and N in coordinated system B respectively.

$\phi_B$ is the degree from $\overrightarrow{MN}\Big|_B$ to X axis.

$N_B$ is the number of robots that have communicated and share the same coordinated system with Robot B.

If $N_A > N_B$ , Robot B will merge its relative coordinate system into Robot A's as follows, and vice versa.

Step 1: Get value of $\phi = \phi_A - \phi_B$ ,

Step 2: Convert the measurements of every point in coordinate system B under the Cartesian system to ones under the polar coordinate system $(X_b, Y_b) \rightarrow (\gamma_b, \phi_b)$

Step 3: Update $\phi_b$ to $\phi_n = \phi + \phi_b$, meaning that coordinated system B rotate $\phi$ degree anticlockwise. (Fig 3a,3b))

Step 4: Convert $(\gamma_b, \phi_n)$ to $(X_n, Y_n)$ under the Cartesian system

Step 5: Add $(M_{xa} - M_{xb}, M_{ya} - M_{yb})$ to $(X_n, Y_n)$ .(Fig 3c,3d)

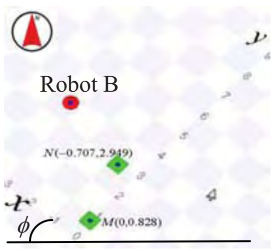

Fig3(a)
$\phi = \phi_A - \phi_B$= 1.0-1.89= -0.89

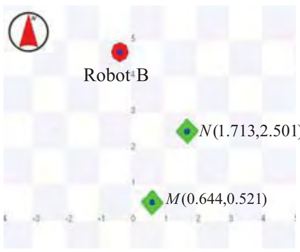

Fig3(b):Coordinate Rotation
Update $\phi_b$ to $\phi_{new} = \phi + \phi_b$

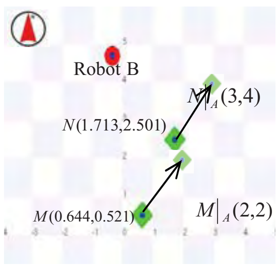

Fig3(c): Coordinate Translation

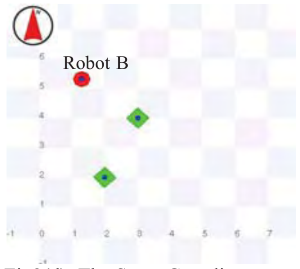

Fig3(d): The Same Coordinate System with Robot A

Now, Robot B and A share the same coordinate system, $N_A$=$N_A$+1, $N_B$=$N_B$-1. M and N served as the reference objects in unifying coordinate system. When robot receives message from another one in the same coordinate system, it will adjust its coordinate system to the average value of them to reduce error. All different relative coordinate systems will merge into a unified one over time. Notice that we are not to build a precise map of the search area. As long as robots share a unified coordinate system, the accuracy of it does not matter very much. We will see that how the amounts of obstacles influence overall system performance in Section V.

## IV. A PSO-INSPIRED DISTRIBUTED COOPERATIVE ALGORITHM

### A. Grid based Map

The map stored in robot's memory is divided into grids of equal size. The length of grid is taken as the unit of the axis. When a robot detects the central position of grid, it will take this grid as "explored". Robot will update its map when it explore a new grid or when receive new grid information from another robot which is in the same coordinate system.

### B. Efficiency

We define robot's "efficiency" as the amount of previous unexplored grids that has been explored by it during the latest T seconds. Fitness value of grid in map is PF. Higher fitness value is roughly equal to more potential unexplored grids around.

$$PF = \begin{cases} E/Speed & 0 < t < T \\ 0 & t > T \end{cases} \qquad (1)$$

E = efficiency of the latest robot which explored this grid.

Speed = speed of the latest robot which explored this grid.

T = time per iteration

t = the interval between be explored by robots.

### C .PSO based on efficiency

As shown in fig 1 (b), without any cue about targets, robot cannot get its local best value which is proportional to the strength of signal from target. We should choose another fitness function to calculate local best. In this situation, the more rate of coverage of the area, the more likely to find the targets, so robots should explore as much new areas as possible.

$$\begin{aligned} V_t &= w * V_{t-1} + L\_F * rand()(X_l[i,j] - X[i,j]) + F \\ X_t &= X_{t-1} + V_t \end{aligned} \qquad (2)$$

$$\begin{aligned} V_t &= w * V_{t-1} + L\_F * rand()(X_l[i,j] - X[i,j]) \\ X_t &= X_{t-1} + V_t \end{aligned} \qquad (3)$$

(2) is the movement model of robot when there are some robots within its neighborhood R. (3) is the movement model when no one else is within R. $V_{t-1}$ is the previous speed. w is the inertia coefficient which slows the velocity over time to prevent explosion of swarm. rand () is a random number between 0 and 1. $X_{t-1}$ is the current position of robot. $X_t$ is the position of the next moment

(a)Local Best: The fitness value of grid is dynamic. Robot compares its maximum of fitness value of grids in its coordinate system with another one's within its neighborhood. And take the maximum value as its local best. $X_l[i, j]$ is the position of local best. Every robot will broadcast its message about its map and local best every T seconds. L_F is a dynamic weight given to the attraction to the local best position (4). It is in direct proportion to the difference between local best's value and the robot's current efficiency. The larger difference is, the stronger attraction is.

$$L\_F = n(l\_v - c\_v) \qquad (4)$$

n is a constant weight factor. l_v is the value of local best . c_v is robot's current efficiency.

(b) Repulsive Force: Local Best may lead robots to be very greedy in terms of the robots' behavior .They are likely to converge on the same grid whose fitness value is highest. This will lower overall system performance. When there are some other robots within its neighborhood R, robot calculates its repulsive force depending on (5). The repulsive force is implemented using a model based upon Coulomb's law of electrostatic force [16].

$$F = k \frac{q_1 q_2}{r^2} \qquad (5)$$

$q_1$ and $q_2$ represent the charges of two particles

r is the distance between them.

F is the resultant force either repels or attracts the particle to another.

k is a positive constant.

Here, $q_1$ and $q_2$ are represented by robot's speed that is always positive. r is the distance between robots. In other word, every robot will exert a positive repulsive force F on the others within its neighborhood.

(c)Obstacle Avoidance: When the laser sensor equipped on robot detect obstacle around it, the robot switch to "Obstacle Avoidance State" and begin to turn at a constant angular speed until the laser sensor on the front detect nothing . Then robot keep going forward for a while (T_avoid).In case that all lasers detect nothing, the robot switch to "PSO State" back in which its speed is determined by the PSO-Inspired algorithm.

In a word, every robot is always attracted by its local best position. It will make robots converge on someplace to exchange information. Then the repulsive force scatters them again. When robot explored the grid of local best again, it would update its fitness value according to (1). An explicit algorithm flow is given as follows

**Algorithm**

```
While(if there are undetected targets)
{
        Mark the detected grids as explored.
        Distinguish and avoid obstacles if they exist.
        Broadcast locally and periodically
        if there are some robots within its neighborhood then
             if they are in the same coordinate system then
                  adjust its coordinate system to the average value
                  move according to (2)
             else
                  if they share at least two reference points then
                       unify their coordinate systems
                       move according to (2)
                  else
                       move according to (3)
        else
             move according to (3)
}
```

## V. EXPERIMENT

We will analyze the scalability and performance of the overall system. Its performance will be compared with a simple algorithm that is not based on accurate global information either. The number of robots and reference obstacles, range of communication R(size of neighborhood) play important roles in our algorithm, so we will analyze these parameters in detail.

### A. Experiment Setup

Experiments were conducted using Player/Stage simulator developed by Brian Gerkey, Richard Vaughan, Andrew Howard, and Nathan Koenig [9].The search area is a square of 40m * 40m which is divided to 1600 grids. Obstacles and robots are initially placed within the area with random position and heading. All the robots are identical. They are equipped with eight IR sensor giving a 360° field of view with a range of 10cm. The camera on robot has a view of 30°. If the distance between robot and target is less than 0.5m, the target can be detected by robot's camera. The speed of robot is bounded to [1.0,2.0] m/s .Any value that is less than 1.0 is considered as 1.0 and any value greater than 2.0 is considered as 2.0.The parameters of algorithm mentioned above were set according to Table I.As an instance ,the process of multi-robot search is illustrated in Fig 4. When a robot finds a target, it will display a pink box containing the number of target.

TABLE I PARAMETERS

| Parameter | value |
|---|---|
| Time per Iteration (T) | 3.0s |
| Inertia coefficient(w) | 0.7 |
| Constant weight factor in local best(n) | 0.5 |
| Repulsive Force weight(k) | 3.0 |
| Time for avoiding obstacle(T_avoid) | 2.0s |

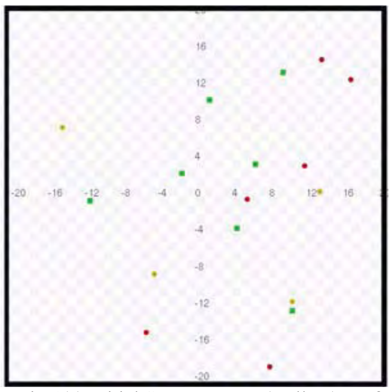
Fig 4(a) Initial State: Targets (Yellow), Reference Points (Green) and Robots (Red) are distributed randomly

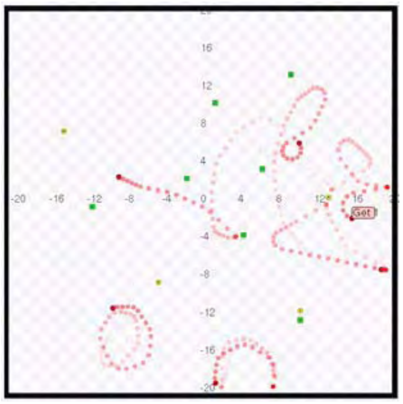

Fig 4(b) Searching

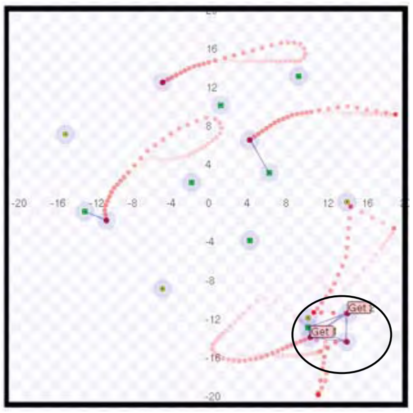

Fig 4(c) Gather for Communication (Robot within the circle)

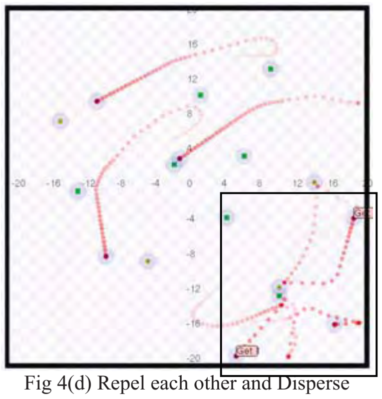
Fig 4(d) Repel each other and Disperse

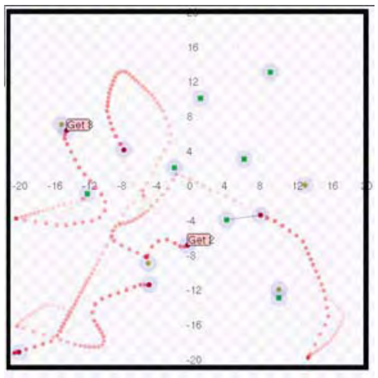
Fig4(e) End

## B. Performance Comparison

Because most of previous work depended on global information, a simple algorithm independent with information on environment is implemented as benchmark: Robots are randomly distributed in the area. Every robot always keeps going straight ahead. When an obstacle is detected in its way, it will adopt the same obstacle avoidance strategy mentioned before. The speed of robot is 1.5m/s (average of 2.0 and 1.0). The performance average over 10 runs for varying numbers of targets can be seen in fig5.The number of robots is 6.

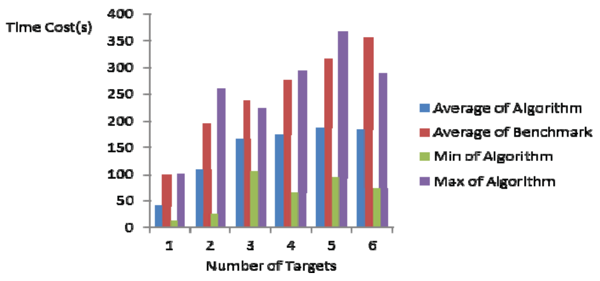

Fig5: Performance of Varying Numbers of Targets

The time cost represents the overall performance. It is obvious that our algorithm outperform the benchmark. We can see that the maximum of time cost in test case with 5 targets is higher than the one with 6 targets. It is because in that test case, targets, reference points and robots are distributed separately as three groups in the area coincidently (as shown in Fig 6).Owing to lack of enough reference points, robots cannot exchange information when they start to search for targets. The lower left corner of the map has been explored repeatedly. So the performance is relatively low.

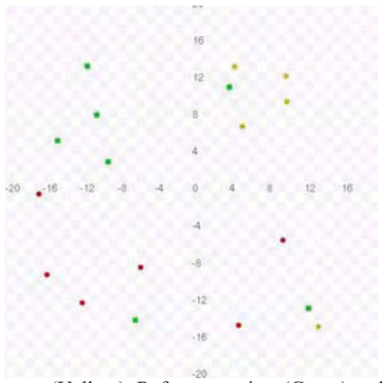
Fig6: Targets (Yellow), Reference points (Green) and Robots (Red) are distributed separately as three groups in the area

## C. Scalability

A remarkable advantage of swarm intelligence is its scalability. The performance keep continuous increases with the number of robots increases. Fig7 illustrates the overall performance of system with different number of robots. There are 4 targets and 7 reference points in the area. We can see from the available statistics that the cooperation among robots reduce time cost in search task.

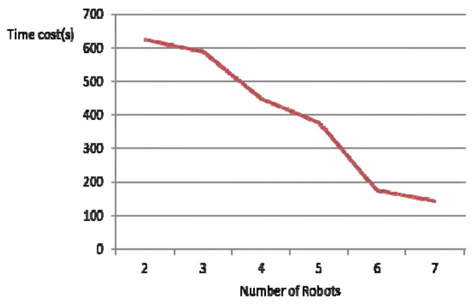

Fig7 Overall Performance of System with Different Number of Robots

## D. The Impact of Obstacles

Sharing at least two reference points are necessary to unify relative coordinate systems for different robots. Here, obstacles do not only stop robots from going straight ahead, but also play the role of reference point. To some extent, more obstacles will increase the overall perform. Figure 8 shows the impact of different numbers of obstacles on performance. The number of robots and targets is 6 and 4 respectively.

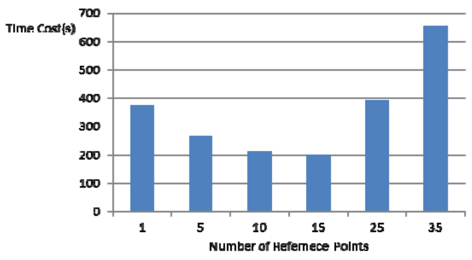

Fig8: Impact of Different Numbers of Obstacles on Performance.

In test case with only one reference point, robots cannot exchange information on the area and targets. There is no concept of cooperation among robots. In test case with 30 reference points, robots spend too much time on avoiding obstacles. So the performance in these cases is lower than the one with 10 reference points.

### E. The Performance with Varying Communication Range

We can see that the performance increase as the range of communication increasing (Fig 9). But the relation between them is not a linear. It is because robots would repel each other all the time if the range of communication is very long. Some targets between two robots are likely to be missed.

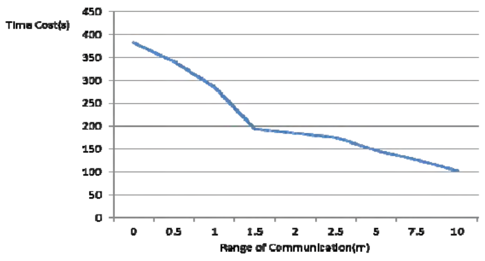

Fig9: Performance with Varying Communication Range

## VI. Conclusion

This paper includes four parts. (1) It analyzes the similarities and differences between swarm particles optimization and multi-robot search. We believe that some ideas in swarm optimization can be borrowed to improve multi-robot search system(2)A method for unifying relative coordinate systems was proposed as the basis for cooperation among robots.(3)A PSO-inspired algorithm based on efficiency was elaborated. The experiment results show that this algorithm can still achieve desired result without global information.(4) The impact of some parameters on overall performance was analyzed , and the result show that communication range R and the number of obstacles has an optimum values within the allowed ranges.

As we mentioned before, communication make a key role in multi-robot system. Due to the limited communication range, we believe that if some robots explored together and form a subgroup, the performance of total system will be improved remarkably. If robot can get accurate information on its location, the algorithm presented here is also suitable for exploring an unknown area to build a map of environment. Furthermore, we have built some real robots shown in fig 10. It is hoped that the work presented here will be implemented on our real robots.

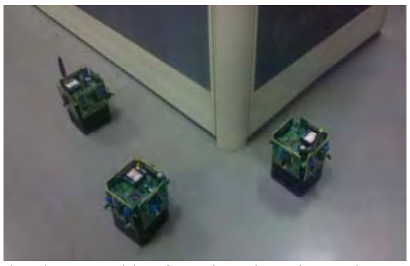

Fig10: Real Robots capable of moving, detection and communication

## References

[1] Acar, E. U., Choset, H., Yangang, Z., & Schervish, M., 2003. "Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods", International Journal of Robotics Research, Vol. 22, No. 7-8, pp. 441-466.

[2] Kantor, G., Singh, S., Peterson, R., Rus, D., Das, A., Kumar, V.Pereira,G., & Spletzer, J., 2003. "Distributed search and rescue with robot and sensor teams", *Proc. of the 4th Intl. Conf. on Field and Service Robotics, Japan*

[3] Landis, G. A., 2003. "Robots and humans: Synergy in planetary exploration", Acta Astronaut, Vol. 55, No. 12, pp. 985-990.

[4] James M. Hereford , Michael Siebold and Shannon Nichols "Using the Particle Swarm Optimization Algorithm for Robotic Search Applications" *Swarm Intelligence Symposium 2007*

[5] Wisnu Jatmiko, Kosuke Sekiyama ,and Toshio Fukuda,"A PSO-based Mobile Sensor Network for Odor Source Localization in Dynamic Environment : Theory, Simulation and Measurement " , *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*

[6] Doctor, S.; Venayagamoorthy, G.K.; Gudise, V.G. "Optimal PSO for collective robotic search applications" *Evolutionary Computation, 2004. CEC2004. Congress on* "

[7] Jim Pugh and Alcherio Martinoli "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization" *Swarm Intelligence Symposium 2007*

[8] Yan Meng and Jing Gan "A Distributed Swarm Intelligence based Algorithm for a Cooperative Multi-Robot Construction Task" *Swarm Intelligence Symposium 2008*

[9] B. Gerkey, R.T. Vaughan, and A. Howard. "The player/stage project: Tools for multi-robot and distributed sensor systems". *In Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), pages 317–323, Coimbra, Portugal, 2003.*

[10] Gerardo Beni "From Swarm Intelligence to Swarm Robotics" *Lecture Notes in Computer Science, 2005, Volume 3342/2005, 1-9.*

[11] J Kennedy, R.Eberhart "Particle Swarm Optimization" *Proceedings,IEEE International Conference on Neural Network,vol.4,pp.1942-1948*

[12] Y. Shi and R. Eberhan,"A modified particle swarm optimizer," *IEEE International Conference on Evolution Computation, 1998, pp. 69 -73.*

[13]James Kennedy Russell C. Eberhart and Yuhui Shi "Swarm Intelligence" page 140-141.

[14]Jim Pugh and Alcherio Martinoli "Distributed Adaptation in Multi-robot Search Using Particle Swarm Optimization" *FROM ANIMALS TO ANIMATS 10 Lecture Notes in Computer Science, 2008, Volume 5040/2008, 393-402*

[15]Derr, K. Manic, M. "Multi-robot, multi-target Particle Swarm Optimization search in noisy wireless environments" *Human System Interactions, 2009. HSI '09. 2nd Conference on*

[16] J. L. Baxter, E. K. Burke, J. M. Garibaldi and M. Norman "Multi-Robot Search and Rescue: A Potential Field Based Approach" *AUTONOMOUS ROBOTS AND AGENTS Studies in Computational Intelligence, 2007, Volume 76/2007, 9-16*