Research Journal of Applied Sciences, Engineering and Technology 9(8): 594-600, 2015
DOI:10.19026/rjaset.9.1443
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 08, 2014 Accepted: July 19, 2014 Published: March 15, 2015

Research Article

A Modified Particle Swarm Optimization on Search Tasking

[1]Mohammad Naim Rastgoo, [1]Bahareh Nakisa and [2]Mohammad Ahmadi
[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia
[2]Faculty of Computing and Technology, Asia Pacific University of Technology and Innovation (APU), Technology Park Malaysia, Bukit Jalil, Kuala Lumpur 57000, Malaysia

Abstract: Recently, more and more researches have been conducted on the multi-robot system by applying bio-inspired algorithms. Particle Swarm Optimization (PSO) is one of the optimization algorithms that model a set of solutions as a swarm of particles that spread in the search space. This algorithm has solved many optimization problems, but has a defect when it is applied on search tasking. As the time progress, the global searching of PSO decreased and it converged on a small region and cannot search the other region, which is causing the premature convergence problem. In this study we have presented a simulated multi-robot search system to overcome the premature convergence problem. Experimental results show that the proposed algorithm has better performance rather than the basic PSO algorithm on the searching task.

Keywords: Multi-robot search system, particle swarm optimization, premature convergence problem

INTRODUCTION

The results of the analysis of the social characteristics of insects and animals have a strong influence in multi-robot system research. Searching and rescue (Kantor *et al*., 2003; Sahin *et al*., 2008) and task allocation (Floreano and Mattiussi, 2008) in multi-robot systems are inspired from the bio-inspired algorithms (e.g., GA, ACO and PSO). One of the most important algorithms in this domain is Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995), which is based on the population stochastic optimization technique that was inspired by social behavior of bird flocking and fish schooling. Doctor *et al*. (2004) present a new version of PSO on the multi-robot search system and shows that PSO algorithm has an acceptable performance in searching task. Pugh and Martinoli (2007) and Pugh *et al*. (2006) presented a new version of PSO on the searching task, which is based on the just local information. This method is tested based on the limited communication ability and the structure of the neighborhood is modified to a close model. Hereford and Siebold (2010) proposed a modification method for the target localization in the two-dimensional search space. Couceiro *et al*. (2011) presented an approach based on the PSO algorithm to explore the search space. Adaptations of PSO have been used for multi-robot odor search in several instances (Jatmiko *et al*., 2006; Marques *et al*., 2006).

PSO algorithm like most of the stochastic search techniques suffers from the Premature Convergence problem. This problem on the multi-robot search system is more significant when the fitness function is limited and each particle just can sense the limited search space around it-self. It means, as the time increase if the particles cannot sense the target in the first iterations then the particles converge to the small regions because of decreasing global searching. Couceiro *et al*. (2011) proposed new method based on the Particle Swarm Optimization (PSO) and Darwinian Particle Swarm Optimization (DPSO) named RPSO and RDPSO. This method is adapted to the multi-robot search systems that take into account the obstacle avoidance. The results showed that the RDPSO increase the search exploration that can avoid being stuck in local optima and can converge sooner to the desired objective value in compare with RPSO.

We present a simple and effective algorithm namely MATRE-PSO on the multi-robot search system. This algorithm by inspiring from the ATRE-PSO (Pant *et al*., 2007) can increase the diversity among the robot and overcome the premature convergence problem.

To test the performance of the algorithm in the realistic system, large quantities of computational time may require. This limitation motivates the use of abstracted models, which uses approximations of details of the system, which have little impact on the targeted performance metrics. Therefore, to validate the effectiveness and usefulness of these algorithms, we

Corresponding Author: Mohammad Naim Rastgoo, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia

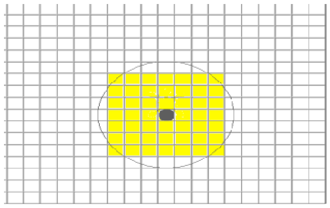


Fig. 1: The specified search space around the robot that can sense

developed a simulation environment for conducting simulation-based experiments in different scenarios and report our experimental results.

METHODOLOGY

Basic Particle Swarm Optimization (PSO) model: In the Particle Swarm Optimization (PSO), which is a new optimization search technique, the particles fly through the multi-dimensional search space to find the potential solution. PSO is initialized with a random number of particles that are spread in the search space to find the optima by updating generations. In the swarm, each particle has a fitness function values in its position and a velocity that guide the particle to move toward the next position. In every iteration, each particle is updating based on the local best solution (p_{best}) that is achieved by each particle so far and the global best solution (g_{best}) that is obtained by the whole of the swarm. After finding the two best values, each particle update its velocity and its position based on the following formula:

$$v_{i,j}(t+1) = \omega.\ v_{i,j}(t) + p\omega.\ rand.\ (p_{best}(t) - x_{i,j}(t)) + n\omega.\ rand.\ (g_{best}(t) - x_{i,j}(t)) \quad (1)$$

$$x_{i,j}(t+1) = v_{i,j}(t+1) + x_{i,j}(t) \quad (2)$$

The first Equation contains three parts: momentum, cognitive component and social component. The inertia weight ω (Shi and Eberhart, 1998) and acceleration constant c_1, c_2 are assumed to be 0.9…. 0.5 and 2 and 2, respectively. r_1, r_2 are the uniformly generated random number in the range of (0, 1). In the beginning, t = 0, $p_{best}(0)$ is the first position of each robot and $g_{best}(0)$ is the first position of the first robot. The termination criteria are also need to be taken into account to get good solution in the acceptable time. In this study, if one of the robots reaches the target or the number of iterations exceeds maximum iterations which are assumed to be 400 iterations the termination criteria occur and the program terminate. If the number of iterations exceeds 400 iterations it means that the algorithm could not find the target. In this study, virtual particles represent the real robots. The real search space is transformed into two-dimensional search space that is divided by squares called cells. The search space of this study contains static obstacles and a target therefore some cells occupied by static obstacles and the robot cannot go to these cells. On the other hand, to prevent the collision between the robots and static obstacles we need to approximate the continuous movement of the robots by dividing the displacement into multiple steps and checking for collisions at each. In this paper we use the method that is introduced by Liu *et al*. (2012) to prevent robots from possible collisions. In this new method each robot generate its route independently and then checks the collision between them. There are separate paths for each robot from the initial position to the goal position. The aim of this method is to find the optimal path, which is the path with the lowest total cost. In this new method each robot replan their route as optimality as possible. In this study each robot can sense the limited cells around its-self and if the target is placed in this area then the fitness function is calculated otherwise is returned zero. Figure 1 shows the number of cells that is sensed by a robot.

Attraction and repulsive of PSO on the multi-robot search system: We proposed an algorithm to overcome the problem of premature convergence. In this method based on the Diversity value (*Div*) there are three phases: attraction, repulsion and combination of attraction and repulsion. When the amount of Diversity is placed above the upper threshold (d_{high}) then the algorithm guide the robots to move toward each other and they move to the next position based on the Eq. (1) as they do in Basic PSO. This coming toward each other causes the gradual decreases in the value of Diversity and this decreases continues until reaches the specific value (d_{low}) then the algorithm switches to the repulsion phase. The repulsion phase guide the robot to move away from the global best position and its own best position achieved so far. In some cases these repulsion phase pushes the robots to move toward one of the corner of the search space and the next position of the robot may place out of the search space due to the search space is limited. To avoid this problem the robot in addition of move away from the global best position and its own best position, they has to move away from its previous velocity direction that defined as:

$$-\omega v_{id} - c_1 r_1 (pbest_{id} - x_{id}) - c_2 r_2 (gbest_{id} - x_{id}) \quad (3)$$

Reversing the velocity direction ($-\omega$) helps the robots to move toward the inside of the search space and escape from the corner of the search space.

The third phase of this method is the combination of the attraction and repulsion phase and when the

```
Program MATRE-PSO

initialization ();

While not done do

        Update MATRE-Velocity ();        //New

        New position ();

        Calculate Fitness function ();   //New

        Calculate Diversity ();
```

Fig. 2: The pseudo code of MATRE-PSO

Function MATRE-Velocity

if Div > d_{high}

$$\omega v_{id} + c_1 r_1 (pbest_{id} - x_{id}) + c_2 r_2 (gbest_{id} - x_{id})$$

Else if d_{high} < Div < d_{high}

$$\omega v_{id} + c_1 r_1 (pbest_{id} - x_{id}) - c_2 r_2 (gbest_{id} - x_{id})$$

Else Div < d_{low}

$$-\omega v_{id} - c_1 r_1 (pbest_{id} - x_{id}) - c_2 r_2 (gbest_{id} - x_{id})$$

Fig. 3: The pseudo code of MATRE-velocity

amount of the diversity lies between the lower threshold (d_{low}) and upper threshold (d_{high}), the robot move toward the its own best position and move away from the global best position.

The pseudo-code for the MATRE-PSO algorithm is shown as follow.

The first of the two new functions (Fig. 2), update MATRE-Velocity, are used for calculating its velocity (Fig. 3). In this part firstly, the diversity measure (*Div*) of the swarm is taken as (Pant *et al*., 2007):

$$diversity(\text{Div}) = \frac{1}{n} \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_x} (x_{ij}(t) - \overline{x_j(t)})^2} \quad (4)$$

where,

- S = The swarm
- n_s = S is the swarm size
- n_x = The problem dimensionality
- x_{ij} = The *j*'s value of the *i*'s robot
- $\overline{x_j(t)}$ = The *j*-dimension average among all robots that is calculated according to the following formula:

$$\overline{x_j(t)} = \frac{\sum_{i=1}^{n_s} x_{ij}(t)}{n_s} \quad (5)$$

The robot based on the Diversity value select one of the three phases and move to the next position.

The values of d_{high} and d_{low} that influence the efficiency of MATRE-PSO, express the upper bound and the lower bound of the diversity of species respectively. The higher values for the d_{high} represents the higher diversity among the robots, so the convergence speed will be lower. Lower value for the d_{low} causes the diversity of the population decrease and the convergence speed increase. So the values of the d_{high} and d_{low} should be neither too low nor too high and we can choose from the experiential values.

RESULTS AND DISCUSSION

The results of the proposed algorithm MATRE-PSO and basic PSO are presented on a group of agents (*i.e.*, robots) that are deployed randomly in the search space. Since both MATRE-PSO and basic PSO are stochastic algorithms, every time they are executed they may lead to different trajectory convergence.

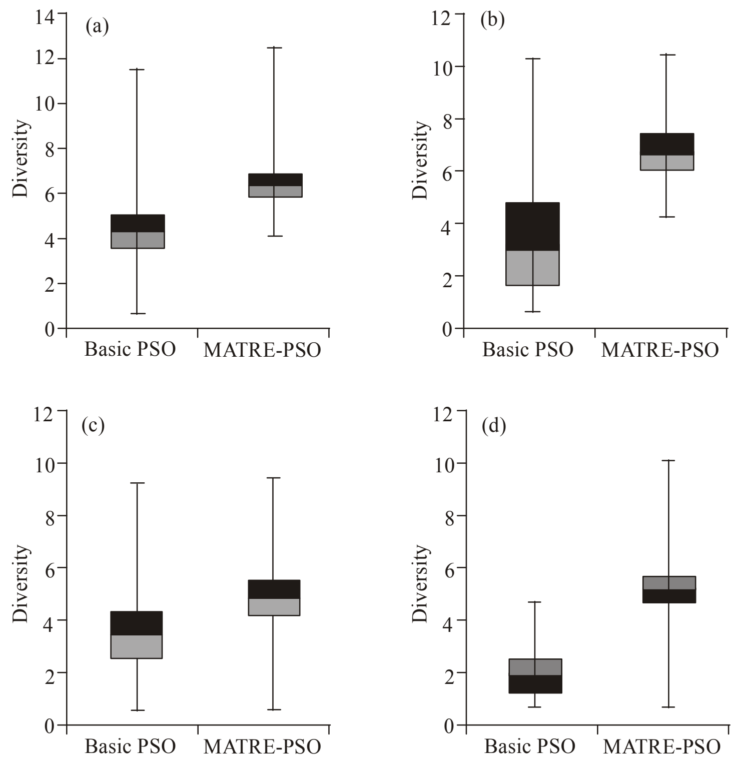


Fig. 4: Diversity among the robots by using MATRE-PSO and basic PSO in different test case, (a) target point 1, (b) target point 2, (c) target point 3, (d) target point 4

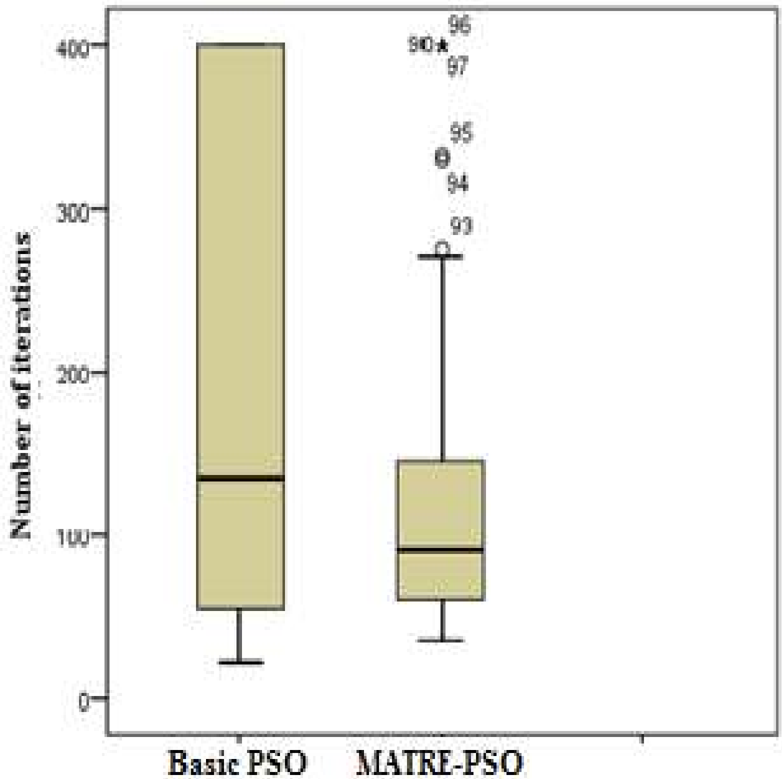
Therefore, multiple test groups of 100 trials of 400 iterations each were considered. The termination criteria met when one of the robot reach the target before 400 iterations or the number of iterations exceeds 400 iterations. Four different positions near the corner of the search space are chosen for the target.

To evaluate the diversity among the robots by applying the MATRE-PSO and basic PSO we made several simulation runs. To make the worst case in each test case, four initial targets and robot position are used. In the other words, in search test case the target is placed in the farthest place toward the initial position of robots and they cannot sense the target in the first iterations. Figure 4 shows the diversity among the robots by applying mentioned algorithm in four different test cases. The diversity of the algorithms was calculated according to the formula (4) and (5).

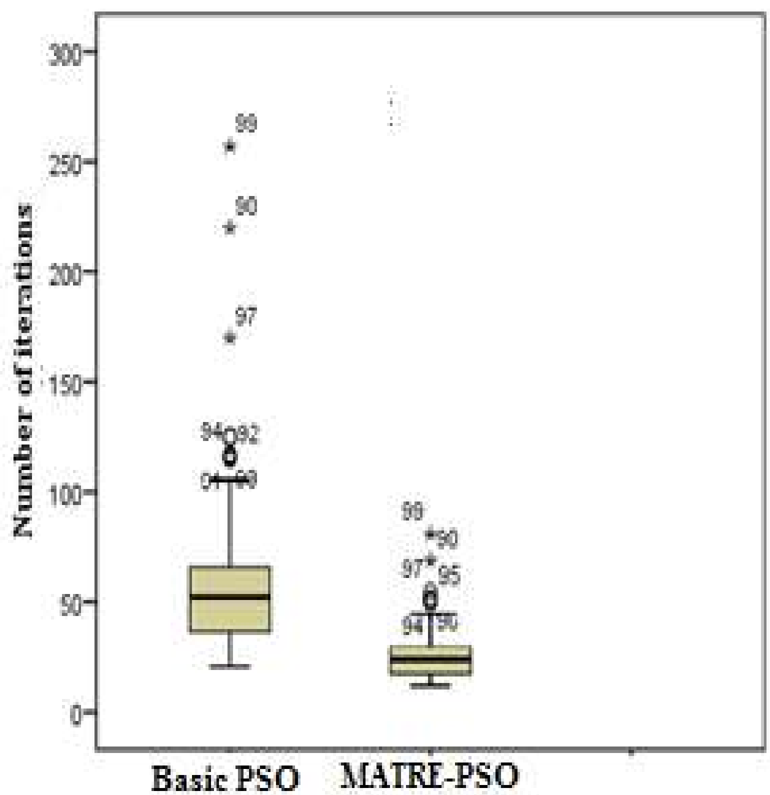
As can be seen from the Fig. 4, the diversity among the robots by applying MATRE-PSO in all test cases is better than Basic PSO. For instance, the diversity of Basic PSO in (b) is between 1.9 and 4.8 whereas the diversity of MATRE-PSO is between 6 and 7.8. It means, due to the lower diversity among the robots, Basic PSO easily entraps into the local optima and they converge to the same regions and cannot search more regions.

In overall, MATRE-PSO has a better diversity among the robots when compared with the Basic PSO in the four experimental datasets. Since these simulation experiments represent a search task, it is necessary to evaluate not only the diversity among the robots but also the search time. Therefore, to further compare both algorithms, the search time of MATRE-PSO and Basis PSO can be analyzed for the worst-case scenarios. In these scenarios, if at least one robot sense the target and reaches it the program terminated. If the robot in the swarm with any algorithms reaches the target in lesser search time then that algorithm will have a better performance. Figure 5 shows the search time of MATRE-PSO and Basic PSO.

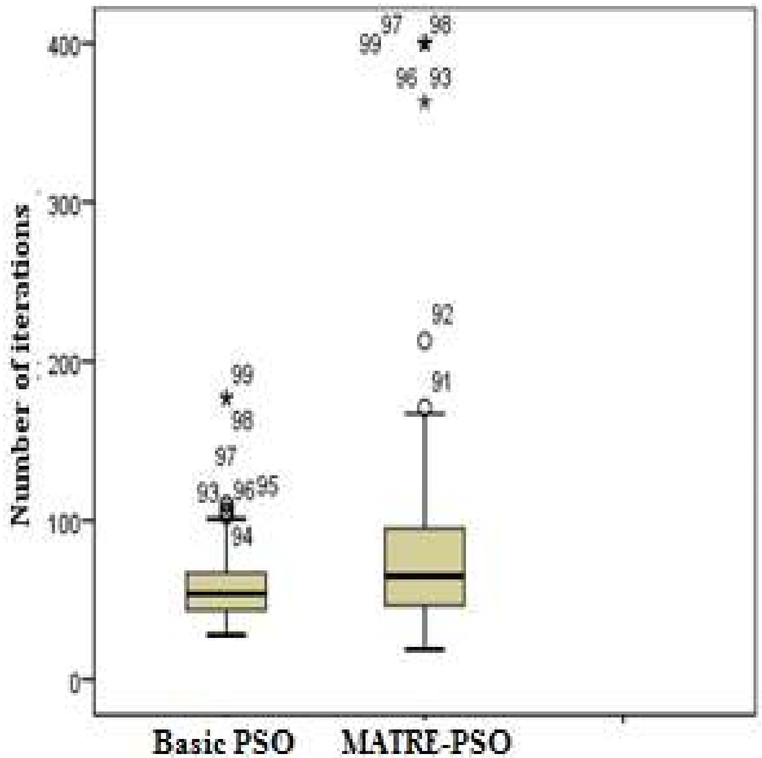
Figure 5 show that MATRE-PSO in overall outperforms the Basic PSO. Although the MATRE-PSO search time in the three of test cases is less than the Basic PSO but in one case (4 (c)) it is more than Basic PSO. In the Fig. 4c the number of iteration in the ATRE-PSO is more than Basic PSO and this is due to the target position that is placed near the initial robots position. In this case the Basic PSO has no difficulties to find the target but the ATRE-PSO because of the controlling the diversity it spent more time in the repulsion or positive conflict phases and did not reach the target easily. Therefore, in this situation the search time in ATRE-PSO is more than the basic PSO.


Number of iterations
400
300
200
100
0
96
97
95
94
93
Basic PSO
MATRE-PSO

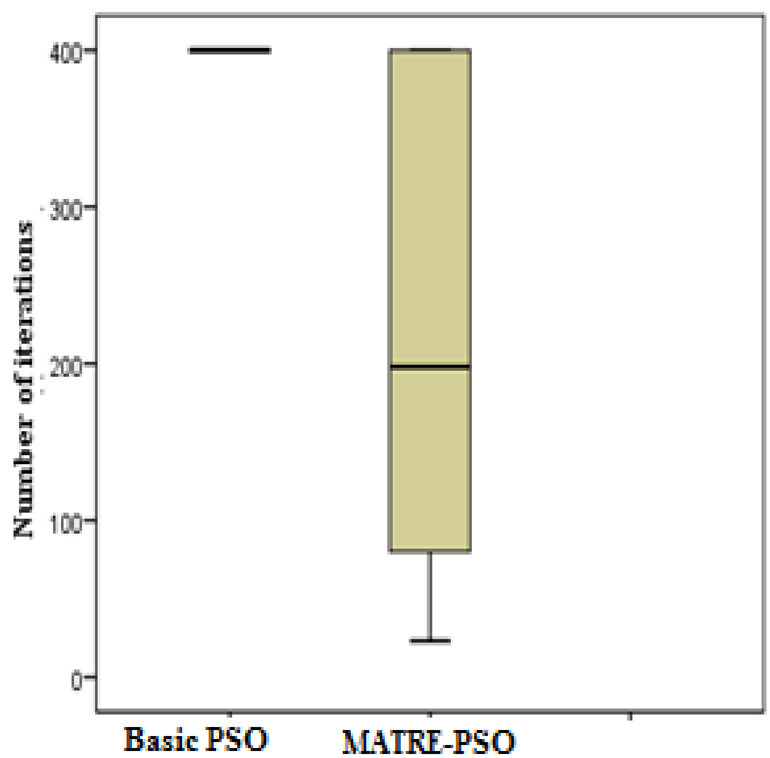
(a)


Number of iterations
300
250
200
150
100
50
0
99
90
97
Basic PSO
MATRE-PSO

(b)



(c)



(d)

Fig. 5: The average search time of MATRE-PSO and basic PSO in different test cases, (a) target point 1, (b) target point 2, (c) target point 3, (d) target point 4

CONCLUSION

Maintaining a high diversity while keeping fast convergence are two contradicting features. Multi-Robot Search Algorithm (MATRE-PSO) while maintaining a high level of diversity, gave a better performance than the Basic PSO in term of the average search time. When the target is placed near the initial position of the robots then the performance of Basic PSO is better than the MATRE-PSO. Because the Basic PSO can easily sense the target and go toward the target but MATRE-PSO shuttles between three phases that may increase the distance between the robots and the target and they cannot reach the target easily. The features presented in this document were implemented in a simulation environment and experimental results show how the performance of MATRE-PSO in the multi-robot search systems is better than the Basic PSO in the environment contains static obstacles. One of the future approaches will be testing the other algorithms that increase the diversity instead of the ATRE-PSO algorithm.

REFERENCES

Couceiro, M.S., R.P. Rocha and N.M. Ferreira, 2011. A novel multi-robot exploration approach based on particle swarm optimization algorithms. Proceeding of the 2011 IEEE International Symposium on Safety Security and Rescue Robotics. Kyoto, Japan.

Doctor, S., G. K. Venayagamoorthy and V.G. Gudise, 2004. Optimal PSO for collective robotic search applications. Proceeding of Congress on the Evolutionary Computation (CEC, 2004).

Eberhart, R. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceeding of the 6th International Symposium on Micro Machine and Human Science (MHS'95).

Floreano, D. and C. Mattiussi, 2008. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies. MIT Press, Cambridge, MA.

Hereford, J.M. and M.A. Siebold, 2010. Bio-inspired search strategies for robot swarms. In: Martin, E.M. (Ed.), Swarm Robotics from Biology to Robotics, InTech 2010.

Jatmiko, W., K. Sekiyamaand T. Fukuda, 2006. A PSO-based mobile sensor network for odor source localization in dynamic environment: Theory, simulation and measurement. Proceeding of the IEEE Congress on Evolutionary Computation (CEC, 2006).

Kantor, G., S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar and J. Spletzer, 2003. Distributed search and rescue with robot and sensor teams. Proceeding of the 4th International Conference on Field and Service Robotics.

Liu, F., A. Narayanan and Q. Bai, 2012. Effective methods for generating collision free paths for multiple robots based on collision type (demonstration). Proceeding of the 11th International Conference on Autonomous Agents and Multiagent Systems, 3: 1459-1460.

Marques, L., U. Nunes and A.T. de Almeida, 2006. Particle swarm-based olfactory guided search. Auton. Robot, 20(3): 277-287.

Pant, M., T. Radha and V.P. Singh, 2007. A simple diversity guided particle swarm optimization. Proceeding of the IEEE Congress on Evolutionary Computation (CEC, 2007).

Pugh, J. and A. Martinoli, 2007. Inspiring and modeling multi-robot search with particle swarm optimization. Proceeding of the IEEE Swarm Intelligence Symposium (SIS, 2007).

Pugh, J., L. Segapelli and A. Martinoli, 2006. Applying Aspects of Multi-robot Search to Particle Swarm Optimization. Proceeding of International Workshop on Ant Colony Optimization and Swarm Intelligence. Springer, Brussels, Belgium, pp: 506-507.

Sahin, C.S., E. Urrea, M.U. Uyar, M. Conner, I. Hokelek *et al*., 2008. Self-deployment of mobile agents in MANETs for military applications. Proceeding of the Army Science Conference, pp: 1-8.

Shi, Y. and R. Eberhart, 1998. A modified particle swarm optimizer. Proceeding of the IEEE World Congress The 1998 IEEE International Conference on Evolutionary Computation, pp: 69-73.