Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MULTI-SWARM PARTICLE SWARM OPTIMIZATION WITH LOCAL SEARCH ON MULTI-ROBOT SEARCH SYSTEM

BAHAREH NAKISA, MOHAMMAD NAIM RASTGOO, MOHAMMAD FAIDZUL NASRUDIN, MOHD ZAKREE AHMAD NAZRI

Department of Computer science, Faculty of Information Technology and Science, University Kebangsaan Malaysia, Bangi, Malaysia

E-mail: Naim.rastgoo@gmail.com, Bahareh.nakisa@gmail.com, mfn@ukm.edu.my, zakree@ukm.edu.my

ABSTRACT

This paper proposes a method based on the Multi-Swarm Particle Swarm Optimization (PSO) with Local Search on the multi-robot search system to find a given target in a Complex environment that contains static obstacles. This method by applying Multi-Swarm with Multi-Best particles on the multi-robot system can overcome the premature convergence problem, which is one of the main problems of Basic PSO. As the time progress the global searching of the algorithm decrease and therefore the robots tend to get group together in the small-explored region that called Premature Convergence and cannot reach the target. By combining the Local Search method with Multi-Swarm, We can guarantee the global convergence of this proposed algorithm and the robots can reach the target. The Experimental results obtained in a simulated environment show that biological and sociological inspiration could be useful to meet the challenges of robotic applications that can be described as optimization problems.

Keywords: *Particle Swarm Optimization; Multi-Swarm And Multi-Best PSO; Premature Convergence, Exploration And Exploitation.*

1. INTRODUCTION

Searching a target in an unknown and dangerous environment is mostly performed by multiple autonomous robots. By applying search algorithm, we can improve the performance of the multi-robot system in terms of more effective exploration and reduce the search time. Therefore there are several advantages in multiple robots compare with single robots such as flexibility, robustness to failure [1].

There is a need for a suitable search task in order to carry out the search task in multi-robot systems fast and effective. These algorithms must be able to make decentralized decision and dynamic neighborhood topology of communicating agents. It means, the robots should be able to operate autonomously without any centralized supervisor.

In 1995 Kennedy & Eberhart introduced Particle Swarm Optimization [2, 3] that is an evolutionary computation algorithm like as Genetic Algorithm. This method uses in many fields such as function optimization, neural network training , fuzzy system control and pattern identification due to the rapid convergence, simplicity and few parameters to be adjusted. Recently, PSO has been applied on the multi-robot search system which has a better performance than other bio-inspired methods. Doctor et.al. [4] proposed the adopted version of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PSO on the multi-robot search system. In this method to perform search task efficiently, the PSO is improved by determining the inertia weight (ω) and coefficients (φ_1, φ_2). Hereford [5] proposed Distributed PSO, which eliminates the central robot for coordinating all the robots movements. This method is scalable to a large number of robots. Physically embedded PSO (PePSO) is introduced by J. Hereford & Siebold [6] that simplified the proposed method in 2006. This method by applying the local calculation and no connection between the robots until better performance is found, improves the performance. The result shows that this method is suitable to find the target by three robots and even more. The limitation of this method is that there are not any obstacles in the environment. Xue et.al. [7] proposed a method to control the multi-robot search systems based on the PSO algorithm to find the target. This method does not consider the volume of robots and there is no obstacle in the environment. Adaptations of PSO have been applied to the multi-robot odor search in several instances ([8, 9]). Pugh et.al. ([10, 11]) proposed based on an adapted version of PSO on the distributed mobile robots. Searching in this method is based on the local information and the structure of neighborhood. In this study, the robots are modeled to have limited communications.

Although Basic PSO has shown that has an excellent performance in solving many problems, it suffers from some drawbacks, particularly when it is placed in an environment with a high density of obstacles. One of the problems is the Premature Convergence, which indicates that the global searching of the PSO algorithm reduces, and after several iterations, the robots converge to a small region that may not have the target. In the early iterations, the velocities of each particle is high due to the high inertia weight; therefore, the global searching and exploration of the robots are more than the local searching and exploitation in these iterations. As the time increases, the inertia value decreases. Then, the robots' velocities decrease, and the robots converge to small regions, with no ability to search other regions to find the target. Couceiro et.al. (2011)[12] proposed a new method based on Darwinian particle swarm optimization (*DPSO*), named RDPSO. This method is adapted to the multi-robot search systems that consider obstacle avoidance. The results demonstrated that RDPSO increases the search exploration that can avoid to be entrapped in the local optima and converge sooner to the desired objective value in comparison with Basic PSO that is applied on the multi-robot search systems.

Another problem of Basic PSO is that it cannot guarantee the global search convergence, which increases the search time. To create an efficient balance between exploration and exploitation in PSO many methods are introduced such as ([13,14]). This problem exists on the multi-robot search systems too. It means, the robots in some cases are placed near the target but have to move based on the velocity and position equation of the PSO, and this equation may guide the robots to move to the position that is placed further from the target. This situation clearly causes the search time to increase, and if there are obstacles near the target, then the best global robot may be entrapped into the local optima.

In this study, we present a simple and effective Multi-Swarm PSO with Local Search named MSL-PSO. In our modified version, a new method is introduced based on the multi-swarm multi-Best [15] to overcome the problem of Premature Convergence and creates an efficient balance between exploration and exploitation and reduces the search time.

2. THEORICAL BACKGROUND

2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a new optimization search technique, which solves the numerical optimization problems. The swarm contains several particles that are specified with position ($x_{i,j}$) and fitness function $f\,(x_{i,j}(t)$. Fitness function is used to evaluate the particle success. The swarm flies through the search space and tries to find the potential solution. Each particle of the swarm spread out in the search space with random position and random velocity ($v_{i,j}$) in the n-dimensional search space. At each iteration, every particle is updated based

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

on the two "best" values. The first best value is the best solution has been achieved by each particle so far that is called *Pbest*. The second best value is the best value (best position with high fitness function) that is found among the swarm, which is called *gbest*. Each particle updates its position and velocity according to the following equation:

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

$$v_{i,j}(t+1) = \omega . v_{i,j}(t) + c_1 . r_1 . \left(pbest_{i,j}(t) - x_{i,j}(t)\right) + c_2 . r_2 . (gbest_{i,j}(t) - x_{i,j}(t)) \qquad (2)$$

$v_{i,j}(t)$ *and* $x_{i,j}(t)$ is the velocity vector and position of *i*-th particle in the *j*-th steps respectively. c_1, c_2 are constants and are known as acceleration coefficients which are equal tp $c_1 = c_2 = 2$. r_1, r_2 are the uniformly generated random numbers in the range of (0,1] and ω is the inertia weight [16]. When t=0, $p_{best}(0)$ is the first position of each robot and $g_{best}(0)$ is assumed to be the first position of the first robot.

2.2 Problem Formulation

In this paper, each particle in the PSO swarm is matched with one robot in the multi-robot search system. The robot is assumed to have access to the map of the search space and have information about its location in the environment. The search space in this study is discretized into non-overlapping cells, which contains static obstacles and a specified target. The size of the target is the same as the cell. Each cell in the search space is marked as an unsafe or safe cell. When the cell is occupied with a robot or static obstacles, this cell is marked as an unsafe cell otherwise is marked as a safe cell. This marking system is used by the robots to have correct path planning and move to the safe cells. For the robot path planning, each robot has to cross from the center of the cell on its path. For a single path, the environment is considered as a static world and the problem is solved by the A* algorithm [17]. Traditional A* method computes the optimal path from the start position to the goal position among the static obstacles but it fails in a dynamic environment.

Each robot in this paper is equipped with a camera that can take a picture from the environment which is considered as a fitness function. The camera can detect 10 surrounding cells from the current position of the robot. The robot applies this camera in 5 different directions and if the robot can find the target in these 5 different directions therefore the number of the target pixels is divided into the whole number of the image pixels that is captured by the camera and this value is considered as the fitness function of that robot. Otherwise the fitness function returns Zero. The fitness function is as follows:

$$0 < fitness\ function = \frac{\sum_{i=1}^{n} po_i}{\sum_{j=1}^{m} p_j} < 1 \qquad (3)$$

Where $po_i = \{po_1, po_2, po_3 \ldots po_n\}$a set of pixels of the target in the image is captured by the camera and $p_j = \{p_1, p_2, p_3 \ldots p_n\}$ is a set of pixels in the image captured by the camera.

The termination criteria are also need to be taken into account to get a good solution in the acceptable time. In this paper, if one of the robots reaches the target or the number of iterations exceeds maximum iterations number which is assumed to be 400 iterations, the termination criteria occurs and the program will be terminated. If the number of iterations exceeds 400 iterations, it means that the algorithm could not succeed to find the target and entraps into the local optima.

3. THE ALGORITHM

From the Basic PSO's equations each of the robot updates its velocity and position based on three items: velocity of the particle, P_{best} and g_{best}. In the first iterations, the robots spread out in the search space randomly and gradually the global searching decreases and all the robots stuck into the local optima. All the information trapped by particles is contained in each position tracked by the particle swarm. But the Basic PSO updates the particle's velocity and position

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

based on the two best values and if the two best values trapped into the local optima therefore the velocity of particles decrease and even be zero. In order to overcome the Premature Convergence problem MSL-PSO is proposed. In MSL-PSO the robot's velocity and position are applies *Multi-pbest* and *multi-gbest* instead of single P_{best} and single g_{best}. By applying multi-best method on the searching task, the robots can escape from the local optima. The proposed algorithm (MSL-PSO) based on the Basic PSO, is divided into seven steps. In the first step, Initialization, each robot is placed randomly in the search space with the random velocity and headings. In the second step, the population is divided into m groups randomly with n robots in each group. In the third step, the fitness function of each robot is evaluated based on Eq. 3. In this step when t=0, $p_{best}(0)$ is the first position of each robot and $g_{best}(0)$ is the first position of the first robot in each group. In the fourth step, if the value of the fitness function is better than the others, which are found so far, it is stored as the best position called p_{best}. The m highest p_{best} is determined and stored for evaluating the new velocity equation. The particle with the closest position to the goal gets the highest value in the fitness function and is stored as g_{best}. In this step, g_{best} for each group is calculated therefore m g_{best} is determined and stored to calculate the next velocity. Then in the fifth step, all the groups combine together and make a group. After that, the next position and velocity of each robot are evaluated based on the Eq. 4, 5 in the next steps. These steps repeat until the termination criteria are met. The termination criteria are also need to be taken into account to get a good solution in the rational time. In this paper, 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 occurs and the program terminate. If the number of iterations exceeds 400 iterations it means that the algorithm could not find the target.

The new equation of robot's velocity and position is changed as follows:

$$v_{i,j}(t+1) = \omega . v_{i,j} + \sum_{k=1}^{m} c_{1,ik} . r_1 (pbest_{ijk}(t) - x_{i,j}(t)) + \sum_{k=1}^{m} c_{2,t} . r_2 (gbest_{jk} - x_{i,j}(t)) \quad (4)$$

$$x_{ij}(t+1) = x_{i,j} + v_{i,j}(t+1) \quad (5)$$

Where $\sum_{k=1}^{m} c_{1,ik} = 2, \sum_{k=1}^{m} c_{2,k} = 2$. The process of MSL-PSO is shown in fig.1

```
Program MSL-PSO
    1. Initialization ();
While (termination criteria in not met)
Do
    2. Divide the population into m groups
       randomly with n robot in each group();
    3. Calculate Fitness function for each robot
       ();
    4. Determine and Store m pbest of each
       particle and m gbest of the population ();
    5. Combine all the groups together ();
    6. Update new Velocity based on Eq.4 ();
    7. Update new Position based on Eq.5 ();
End
```

Figure.1. The pseudo code of the MSL-PSO algorithm

4. MSL-PSO RESULT

4.1 Simulation Conditions

We have simulated and tested MSL-PSO and Basic PSO in four different target positions. To show the performance of the proposed algorithm and Basic PSO we put the target in the worst place, which are near or behind the obstacles. Only one target position was activated during each search run.

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

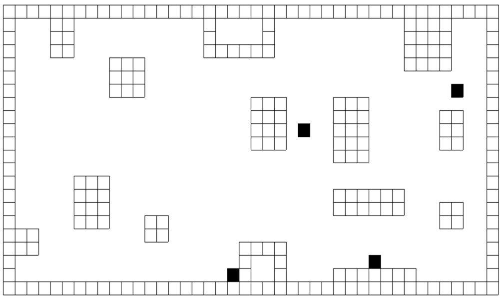


Figure 3. Map of simulation search space and showing the 4 different target point locations.

In this study, the search space in bounded with borders and the robots in this search space cannot go outside of the search space. It means, if the next position of the robot is placed out of search space, the next position would reverse and be placed inside the search space.

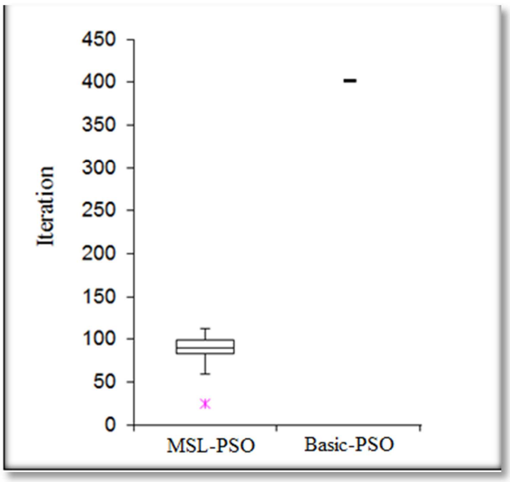
In order to get results, we applied the following parameter values. Inertia coefficient, ω, was set to 0.9…. 0.5. The c_1, c_2 coefficients were both set to 2. We set an initial v for each robot to simulate the behavior of physical robot. In this study only nine robots is used and therefore the population is divided into 3 groups, which contain 3 robots in each group. In each group the g_{best} is calculated separately and then stored for evaluating the new velocity. In each group the g_{best} is calculated separately and then stored for evaluating the new velocity.

In this study we adapted the PSO algorithm on the multi-robot search system, therefore unlike most of the PSO researches that track function value, our simulation search the target function. The simulation stopped when the robot got near the target or the maximum number of iterations (400 iterations) elapsed.

4.2 simulation result

To evaluate the effectiveness of the MSL-PSO is tested in four different target positions and compare with Basic PSO algorithm. To compare the performance of the algorithm the search time is chosen as measurement. For each target positions both algorithms is run 100 test cases and the initial position of each robot is different. The initial positions of robots with each target position changed and are placed in the worst place. It means when the target is placed in some location then the initial positions of robots are selected as a furthest place than the target. Therefore in the first iterations the robots cannot see the target and then the global searching (exploration) of them decrease and they converge to the small regions and could not find the target. In addition, there are two random values (r_1, r_2) that are selected randomly in each test case. The maximum number of iterations for this study is considered 400 iterations. This number is selected because in most cases if the robots cannot reach the target before 400 iterations it means they stuck into the local optima and never reach the target after 400 iterations. Since our main objective is to decrease the search time therefore 400 iterations may be suitable.

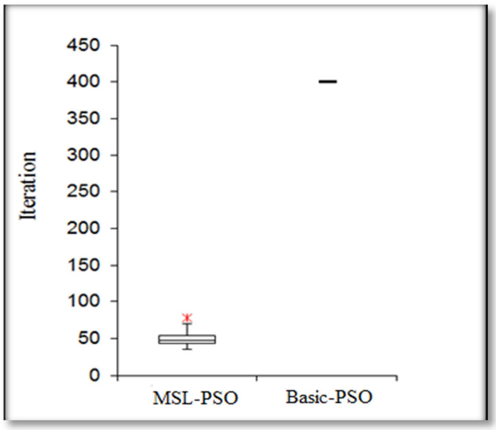
Figure 2 compares the performance of MSL-PSO with Basic PSO in 4 different target positions. The figures show search time (number of iterations passed) for both MSL-PSO and Basic PSO algorithm.



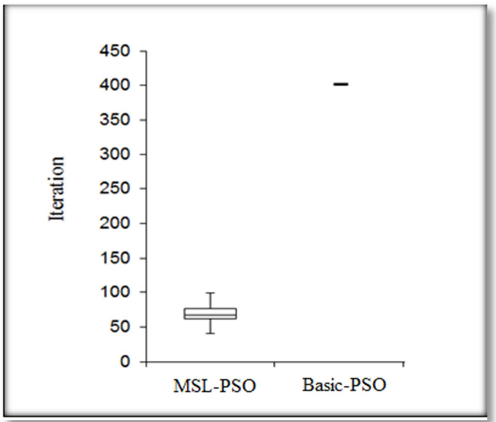
(a)

Journal of Theoretical and Applied Information Technology
10th January 2015. Vol.71 No.1

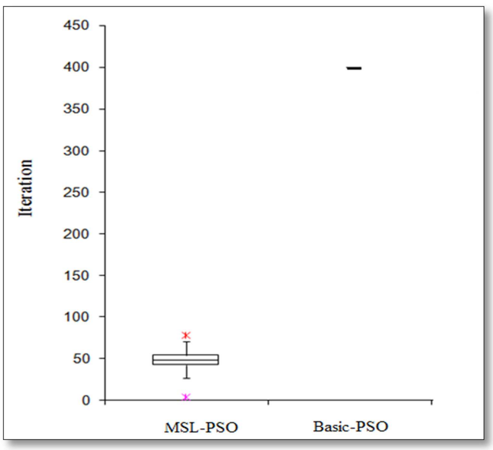
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(b)



(c)



(d)

Figure 5. The result of the search time of MSL-PSO and Basic PSO algorithm with different target positions: (a) Target position 1. (b) Target position 2. (c) Target position 3. (d) Target position 4.

As can be seen in Figure 5, Basic PSO algorithm cannot find the target during 400 iterations in all four different target positions while the robots by applying the MSL-PSO algorithm can find the target and reach it.

In target point 1, The robots through MSL-PSO algorithm can find the target and reach it between 60-120 iterations. In this case the although, the initial positions of the robots are far from the target but by increasing the global searching can search different parts of the environment and can find the target during the desired search time. In this case the robots through the Basic PSO are stuck into the local optima and during the search time search the same region; therefore, they could not find the target never.

As can be seen from Figure 5(b) the number of iterations of MSL-PSO increases. In fact, in this case the initial positions of robots are placed in the farthest position towards the target. In addition to this, the number of static obstacles on the way of the robots during search is high; therefore, they could not see the target in the first iterations and they get entrapped into the local optima. The robots through the Basic PSO could not search the other areas and never find the target in 400 iterations while the robots by applying the MSL-PSO can find the target and reach it between 20-70 iterations.

In target point 3, the target is placed behind the target; therefore, the robots must come near to the target to see it. In this case the robots through the Basic PSO easily trapped into the local optima and could not search the other areas. These robots could not find the target during 400 iterations. In contrast, the global searching of robots by using MSL-PSO increased and they escape from the local optima and search different part of the search space; therefore, they see the target and move towards it through the local search strategy. The number of passed iterations of MSL-PSO is around 40-90 iterations.

As can be seen from the figure 5 (d), the target is placed near the corner and this is the hardest case among all. In the first iterations, the robots are placed in the farthest position to simulate the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

worst case and show the performance of both algorithms better. The MSL-PSO in this situation could find the target between 20-80 iterations while the Basic PSO got entrapped into the local optima and therefore could not find the target during 400 iterations.

5. CONCLUSION

A biologically inspired search strategy on multi-robot search system has been developed and tested. This search technique, which is named MSL-PSO, is based on the bird flocking and Basic PSO and able to solve two problems of Basic PSO.one of these problems id Premature Convergence that is appeared in the environments contain static obstacles. This problem is more evident when the initial position of the robots is far from the target and they are not able to reach it easily. It means, in the first iterations, the global searching of robots are high and they can search the different part of the environment more but as time increase the global searching decrease; therefore, they converge to a region and could not search different parts and may fail to find the target.

Other problem of Basic PSO is that there is no balance between exploration and exploitation. In the other words, in some cases when the robot observe the target and its fitness function value is evaluated high but Basic PSO algorithm guides the robot to move toward the position that increase the distance between the robot and the desired target which increase the search time.

The MSL-PSO increases the global searching (exploration) of the Basic PSO by dividing the population into the *m* groups and evaluating each robot based on the new velocity and position equations (Eq.4, 5).

In order to decrease the search time and create an efficient balance between exploration and exploitation a Local search algorithm is added to the proposed algorithm. In this method when the robot observes the target by applying Local search algorithm move toward the target and can guarantee to reach it.

The performance of The MSL-PSO algorithm and the Basic PSO are tested on the multi-robot search system in the environment with number of static obstacles with considering the target in four different positions. The results show that MSL-PSO has a better performance in compare with Basic PSO in this domain. The MSL-PSO by increase the global searching among the robots guides the robots to move to different parts of the search space; therefore, the robots can find the target and reach it sooner. the results show that the robots through the Basic PSO easily trapped into the local optima and could not find the target during 400 iterations.

REFERENCES

[1] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics: Stateof- the-art Survey*, ser. Lecture Notes in Computer Science (LNCS 3342), E. Sahin and W. Spears, Eds. Berlin Heidelberg: Springer-Verlag, 2005, pp. 10–20.

[2] Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. Paper presented at the Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on.

[3] Kennedy, J., & Eberhart, R. 1995. Particle swarm optimization. Paper presented at the Neural Networks, 1995. Proceedings., IEEE International Conference on.

[4] Doctor, S., Venayagamoorthy, G. K., & Gudise, V. G. (2004). *Optimal PSO for collective robotic search applications.* Paper presented at the Evolutionary Computation, 2004. CEC2004. Congress on.

[5] Hereford, J. M. (2006). *A distributed particle swarm optimization algorithm for swarm robotic applications.* Paper presented at the Evolutionary Computation, 2006. CEC 2006. IEEE Congress on.

[6] Hereford, J., & Siebold, M., 2008. Multi-Robot Search Using A Physically-Embedded Particle Swarm Optimization. International Journal of Computational Intelligence Research, 4(2), 197-209.

[7] Xue, S., Zhang, J., & Zeng, J., 2009. Parallel Asynchronous Control Strategy For Target Search With Swarm Robots. International Journal of Bio-Inspired Computation, 1(3), 151-163.

[8] Jatmiko, W., Sekiyama, K., & Fukuda, T. 2006. *A pso-based mobile sensor network for odor source localization in dynamic environment: Theory, simulation and measurement. Paper presented at the Evolutionary Computation, 2006. CEC*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2006. IEEE Congress on.

[9] Marques, L., Nunes, U., & de Almeida, A.T. 2006. *Particle swarm-based olfactory guided search. Autonomous Robots, 20(3), 277-287.*

[10] Pugh, J., Segapelli, L., & Martinoli, A. (2006). Applying aspects of multi-robot search to particle swarm optimization *Ant Colony Optimization and Swarm Intelligence* (pp. 506-507): Springer.

[11] Pugh, J., & Martinoli, A., 2007. Inspiring And Modeling Multi-Robot Search With Particle Swarm Optimization. Paper presented at the Swarm Intelligence Symposium, 2007. SIS 2007. IEEE.

[12] Couceiro, M.S., Rocha, R.P., & Ferreira, N.M. (2011). A novel multi-robot exploration approach based on particle swarm optimization algorithms. Paper presented at the Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on.

[13] Robinson, J., Sinton, S., & Rahmat-Samii, Y., 2002. Particle Swarm, Genetic Algorithm, And Their Hybrids: Optimization Of A Profiled Corrugated Horn Antenna. Paper presented at the Antennas and Propagation Society International Symposium, 2002. IEEE.

[14] Vesterstrom, J. S., Riget, J., & Krink, T., 2002. *Division Of Labor In Particle Swarm Optimisation.* Paper presented at the Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on.

[15] Li, J., & Xiao, X. (2008). *Multi-swarm and multi-best particle swarm optimization algorithm.* Paper presented at the Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on.

[16] Shi, Y., & Eberhart, R. (1998). *A modified particle swarm optimizer.* Paper presented at the Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on.

[17] Hart, P. E., Nilsson, N. J., & Raphael, B., 1968. A Formal Basis For The Heuristic Determination Of Minimum Cost Paths. Systems Science and Cybernetics, IEEE Transactions on, 4(2), 100-107.