Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

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

TARGET SEARCHING IN UNKNOWN ENVIRONMENT OF MULTI-ROBOT SYSTEM USING A HYBRID PARTICLE SWARM OPTIMIZATION

[1] BAHAREH NAKISA, [2] MOHAMMAD NAIM RASTGOO, [3] MOHD ZAKREE AHMAD NAZRI, [4]MD. JAN NORDIN

[1,2] Science and Engineering Faculty, Queensland University of Technology, Brisbane, QLD, Australia

[3,4] Center for Artificial Intelligence Technology, Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, Malaysia

E-mail: [1]bahareh.nakisa@qut.edu.au, [2]Mohammadnaim.rastgoo@qut.edu.au ,[3] Zakree@ukm.edu.my , [4]Jan@ukm.edu.my

ABSTRACT

Target searching in unknown environment using multi-robot search systems has received increasing attention in recent years. Particle Swarm Optimization (PSO) has applied successfully on multi-robot target searching system. However, this algorithm suffer from premature convergence problem and cannot escape from the local optima. It is, therefore, important to have an efficient method to escape from the local optima and create and efficient balance between exploitation and exploration. In this study, we propose a new method based on PSO algorithm (ATREL-PSO) to find the target in unknown environment using multi-robot system within a limited time. This novel algorithm is demonstrated to escape from the local optima and create an efficient balance between exploration and exploitation to reach the target faster. The concept of attraction, repulsion and the combination of repulsion and attraction enhancing the search exploration, and when the robot get closer to the target it should forget the PSO concept and apply the local search method to reach the target faster. 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: *Swarm Robots, Particle Swarm Optimization, Premature Convergence, Target Searching*

1. INTRODUCTION

Finding target in unknown environments using multi robot search system is one of the important problems in mobile robot research field. For search application, mobile robots are used in many different scenarios such as foraging tasks [1]–[3], search and rescue victims in disastrous environment [4] and firefighting [5]. Using a multi-robot system in searching task can offer several major benefits over the single robot alternative. Searching can be done massively in parallel, significantly decreasing the time taken to locate targets and improving robustness against failure of single agent by redundancy as well as individual simplicity. There are several algorithms inspired by biological societies, which are applied on multi-robot systems. One of the well-known algorithm for multi-robot searching problem is Particle Swarm Optimization (PSO) [6], [7]. PSO algorithm [8], which is based on population stochastic optimization technique, is inspired by social behavior of bird flocking and fish schooling. The first version of PSO on multi-robot search system is proposed by Doctor et al. [3]to find one or multi target cases. In this method PSO algorithm is improved by determining the optimal parameters like inertia weight (ω) and upper bounds of learning coefficients (φ_1, φ_2) to perform the search task efficiently. Hereford [6] proposed an algorithm called Distributed PSO. In this method, each robot calculates its new position and eliminates the central robot to coordinate all robots movements. The result showed that this method is scalable for a large number of robots. Hereford J. and Siebold [7] introduced and simplified a method called physically

embedded PSO. In this method, local robots do all the calculations and there is no communication until a better position is found during the search. The result showed that multi-robot system with three robots can find the target successfully and this method is scalable even if the number of robots increase. The limitations of this method are that the rotation of robots are very restricted and there is no obstacle in the environment. Xue, Zhang, and Zeng [9] Modeled and controlled a multi-robot system parallel-based PSO algorithm to find a target. This method did not consider the volume of robot and there is no obstacle in the environment. Adaptation of PSO has been used for multi-robot odor search in several instances [10], [11]. An adapted version of PSO on a distributed mobile robots to search just based on local information is introduced and the Performance of the algorithm is evaluated when the neighborhood structure was modified to a model with a limited communication abilities [12], [13].
Although PSO has shown a good performance on solving many problems, it suffers from premature convergence and it traps into the local optima. This problem is a common problem among all stochastic algorithms. As the time progresses, global searching of PSO algorithm reduces and after several iterations this algorithm converges to a small region that may not be the target in that region that affects the performance. Although the convergence is a desired property, it may cause the algorithm traps into the local optima and not able to explore the other regions to find the target. Nakisa, Nazri, Rastgoo and Abdullah[14] presented a survey of PSO-based algorithms that solved Premature Convergence problem in different domains. Premature convergence problem in multi-robot system further appeared when the static obstacles are taken into account [15]–[20]. A new method based on the Particle Swarm Optimization (PSO) and Darwinian Particle Swarm Optimization (*DPSO*) named RPSO and RDPSO is presented by Couceiro, et.al [21]. This method, which is adapted to multi-robot search systems, takes into account an obstacle avoidance approach. The result showed that RDPSO increases the search exploration and can avoid the robots being stuck into the local optima and can converge sooner to the desired object in compare with RPSO.

It has been proved that Basic PSO cannot guarantee global search convergence, which increases the search time. In order to improve the algorithm convergence, many scientists introduced different methods by hybridizing PSO to create an efficient balance between exploration and exploitation [15], [16], [18]–[20]. In this article we consider a system consisting of multiple robots deployed in a search space using Particle Swarm Optimization to maintain high-level diversity and global convergence. In this paper, robots can escape from the local optima faster and get the target by applying the local search method (A*). To evaluate the performance of the proposed algorithm in the realistic system, large quantities of computational time may require. This limitation motivates the use of abstracted model, which uses approximations of details of the system, and have a little impact on the targeted performance metrics. Therefore, to validate the effectiveness and usefulness of the algorithms, we developed a simulation environment for conducting simulation-based experiments in different scenarios and report our experimental results.

The reminder of this paper is organized as follows: Section 2 briefly introduces PSO algorithm on multi-robot search system; Section 3 presents our new proposed method (ATREL-PSO algorithm) for searching a target in unknown environment; Section 4 describes the simulation environment and some simulation-based experimental results; Section 5 concludes the paper and discusses future work.

2. PROBLEM FORMULATION

Particle Swarm Optimization (PSO) is a new optimization search technique, which solves the numerical optimization problems [22] . Particles fly through the multidimensional search space to find the potential solution. In the swarm every particle are specified with position $(x_{i,j})$ and performance $f\,(x_{i,j}(t))$ at each iteration $t \in \mathrm{N}$. In each step of the algorithm, an objective function is used to evaluate the particle success. *PSO* thrives to minimize a cost function, or maximize a fitness function. To model the swarm, each particle start to search with a randomized position in the n-dimensional search space with (possibly) randomized velocity $(v_{i,j})$, where $x_{i,j}$ represents the location of particle index *i* in the *j-th* dimension of the search space. The next position vector $x_{i,j}(t+1)$ and the next velocity vector $v_{i,j}(t+1)$ of each particle are highly dependent on the current position vector $x_{i,j}(t)$, velocity vector $v_{i,j}(t)$, local best vector $p_{best}(t)$ and global best vector $g_{best}(t)$ information. Candidate solutions by flying the particles through the virtual space are optimized, with attraction to best positions in the space with the best result. At each time step the velocity is updated and the robots move to the new position that is calculated by the previous position and the new velocity as follow:

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

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

The velocity of each robot is updating by the following formula:

$$v_{i,j}(t+1) = \omega \times v_{i,j}(t) + p\omega \times rand \times \left(p_{best}(t) - x_{i,j}(t)\right) + n\omega \times rand \times (g_{best}(t) - x_{i,j}(t)) \quad (2)$$

Where the inertia weight ω [23] and acceleration constant c_1, c_2 are assumed to be 0.9…. 0.5 and 2 and 2 respectively and 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 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 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 paper multi-robot search using a one-to-one matching between particles in the PSO swarm and robots in the multi-robot system motivates algorithm. We initially assume they have complete knowledge about their location in the environment by accessing to the map of the search space. There are some key differences between PSO and PSO in multi-robot search that require us to make some modifications to the algorithm.

Search space: We have transformed a real space into two-dimensional search space and divided it into squares called cells. Each cells in search space represents a square in the real world with a selected size (for the algorithm itself, the size does not play any important role). The environment in this paper contains static obstacles and a single target. To prevent the collision between the robot and the static obstacles and other robots, the environment should be discretized into the cells and the robot should move into the safe regions. Each cell, which is occupied by the obstacles or other robots, will be marked as unsafe cells. The center of each cell is considered as a point of Interest. It means that if the robot visits the center of the cell, the entire cell is considered as a visited cell.

Robot: In this thesis we assume the geometrical shape of the robot is like a circle with the determined radius (Ŗ) and has the same size as a cell. The state of each robot in the search space is represented by six variables (x, y, v, θ_r, θ_c, t) that are the position of the robot in the 2-D dimensional search space, speed of the robot, head of the robot, the determined direction of the robot to move to the next position and time in that position respectively. The robot is supposed to move toward 8 different directions (θ_c) therefore the robot can move to the adjacent cells (green cells) around its current position (Figure 2). As described the search space is discretized and therefore the path planning of the robot from its current cell to the goal cell is also discretized and the robot must cross through the center of the cells on its route. For a single path the environment is considered as a static world and the problem is solved by the A* algorithm [24]. 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.

Movement limitation: In PSO the particles do not have the limited acceleration and velocity. Due to the Robots exist in the real world, they have limitation to how quickly can move and adjust their headings. In this paper, the robot velocity is discretized into discrete values that enable it to execute just one action at each time step. As illustrated before, there is a limitation in the velocity of the robot and the velocity is placed between $[-V_{max}, V_{max}]$ where the v_{max} represent the maximum velocity of the robot along its direction and $-v_{max}$ is the maximum velocity of the robot but in the reverse direction. If the velocity of the robot is placed out of this range we set this velocity as a Maximum velocity value in each side.

Fitness function: we assume each robot has a camera to capture the picture from the environment. When the robot uses the camera to find the target, if the target is placed in the range of view of the camera then evaluates the fitness function otherwise it returns zero. The fitness function in this study is as follows:

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

Where $po = \{po_1, po_2, po_3 \dots po_n\}$ is a set of pixels of the target in the image captured by the camera and $p = \{p_1, p_2, p_3 \dots p_n\}$ is a set of pixels in the image captured by the camera. It should be noted that the amount of captured pixel of the target is less than the whole image. Therefore, the value of fitness function in this study is in the range [0, 1]. When the position of robot is close to the target, the ratio of captured pixel of target to the whole target is higher that result in high fitness function. However, if the robot is far from the target, then the ratio of target pixel to the whole image is lower which result in lower fitness function. If the robot cannot capture the target the fitness function is zero.

In this study, the robot is able to use their cameras in 8 different directions. Therefore, it has the ability to observe the entire environment by rotating it

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

camera. When the robot stand in one cell we assume that the robot can rotates and takes pictures in 5 directions. The figure 1 shows the 5 directions of the robot in the current position and its adjacent regions.

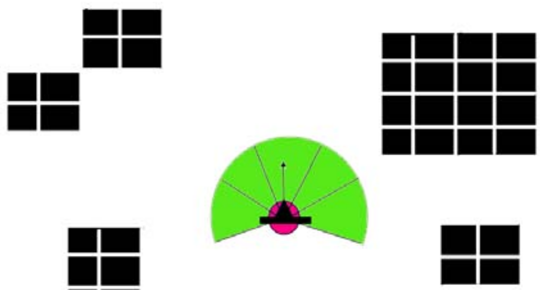
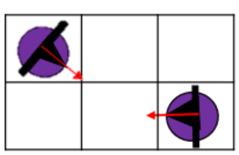
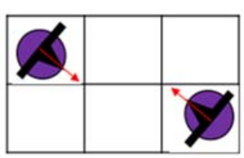


Figure 1: The simulated robot and it five regions and directions

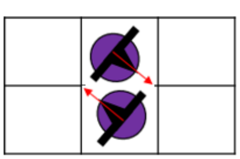
Robot Collision: Using the standard PSO particle displacement at each iteration, we will be unable to detect any collisions that might occur along the path. We therefore need to approximate the continuous movement of the robots by dividing the displacement into multiple steps and checking for collisions at each. In multi-robot system, robots and the target have some volume therefore they have to prevent to collide with each other or static obstacles. In this paper we use the method that is introduced by Liu et.al.[25] 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.



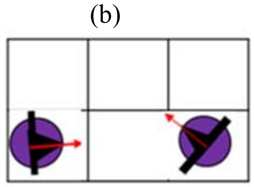
(a)



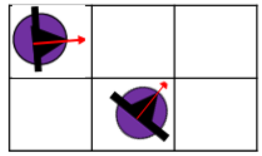
(b)



(c)



(d)



(e)

Figure 2: Illustration of 5 collision types. (a) Head-On. (b) Front Sideswipe. (c) Rear Sideswipe. (d) Front-End Swipe. (e) Front-End Sideswipe.

Robots Communication: In this study, we consider a central station to gather all the information of the robots in the search space and broadcast when it is needed. The central station update the map of the search space based on the current situation of the robots. In this case the central station marked the current position of each robot as occupied cell. This information is updated at each iteration that robots move from one position to another position. This information help to avoid collision with other robots. In addition to the having the current position of the robots, the central station calculated the next velocity and next position of each robot and send them to robots to move to the calculated positions.

3. THE PROPOSED METHOD

3.1 Attraction And Repulsive Of PSO with Local Search

To overcome the problem of premature convergence on the multi-robot search system, we proposed an algorithm, which is proposed by Pant et al. [26]. In this method, depending on the diversity measurement (*Div*), there are three phases namely: attraction, repulsion and combination of attraction and repulsion. In the attraction phase if the amount of diversity is above the upper threshold (d_{high}), the robots move toward each other based on the following Eq. (1) as they do in Basic PSO. This coming toward each other causes the gradual decrease in diversity of the population and this decrease continues until it reaches below the certain value (d_{low}) then switches to the repulsion phase. In the repulsion phase, robots just move away from the global best position and its own best position seen so far to increases the diversity. In some cases, this repulsion phase pushes the robots to move toward one of the corners of the search space. Due to the limitation of search space, the next position of the robot will may place in out of the search space and stuck in the corner of search space. 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 as well that defined as:

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

$$v_{id} = -\omega \times v_{id} - c_1 \times r_1 \times (pbest_{id} - x_{id}) - c_2 \times r_2 \times (gbest_{id} - x_{id}) \quad (4)$$

Reversing the velocity direction $(-v)$ helps the robots to move toward the inside of the search space and escape from the corner of the search space. In the third phase of this method, which is the combination of the attraction and repulsion phase, when the amount of the diversity lies between the lower threshold (d_{low}) and upper threshold (d_{high}), the robot move toward its own best position and move away from the global best position.

On the other hand, in order to guarantee the global convergence of the algorithm when the fitness function of each robot reach higher than the specific threshold, which is specified experimentally, then that specific robot go toward the target by the Local Search algorithm instead of the ATRE-PSO. This paper uses the A* [24] algorithm as the Local Search method and the value of the threshold is given experimentally that is different for different environments. In the A* starts from the current node and continues until reaching the determined lookahead that is equal to 1 for this study [27]. As described before, there are 5 adjacent cells around the current position of the robot that it can move to them by the specific direction. When the camera of the robot rotates, it can evaluate the fitness function for each 5 directions. Then the A* algorithm by selecting the largest f-value that is belong to the specific direction, move toward the adjacent cell along this specific direction. The f-value for these directions is calculated by the following formula:

$$f(n) = g(n) + h\,(n) \quad (5)$$

The *h(n)* is the cost-to-go, which is the fitness function value of the robot current position in the specific direction. *g(n)* is the cost-thus-far that is the cost from the current node to the next position and due to the lookahead is equal to one then the g (n) in this study is equal to one. In each step, A* by starting from the current position in the search space until reaching the specific lookahead selects the states and this search does not finish during the lookahead steps until reaching the goal. This chosen states form the local search space. There are two lists in this algorithm named: Open and Close. The Open is the list that stores all the acceptable directions of the robot, which has the specific fitness function value and then sorts them. The sorting of the Open list is based on the Max- Heap in this study and while each direction is added to the Open list, the list is reordered based on the biggest f-value. It means the top of the list refers to the biggest f-value. The selected direction with the biggest f-value pops up from the Open and is put in the Close. Then the algorithm selects a state from the neighbor of the current state of the robot and guides the robot to move to the state with the best fitness function value. The pseudo-code for the ATREL-PSO algorithm is shown as follow:

Algorithm 1. The pseudo code for ATREL-PSO

Start

Create and initialize swarm with N robots
//each particle is the same as a particle in PSO
Initialize variable and parameters
Calculate the first position randomly;
While (stopping criteria in not satisfied) do
    For each robot i=1, 2,…, N do
        //Set the best personal for each robot
        Calculate the Diversity (*Div*) using Eq.4
        If fitness function $(robot_i, position_{robot_i}) > \theta$
            A*-Algorithm using Eq. 5
        Else if $Div > d_{high}$
            $v_{id} = \omega \times v_{id} + c_1 \times r_1 \times (pbest_{id} - x_{id}) + c_2 \times r_2 \times (gbest_{id} - x_{id})$
        Else if $d_{high} < Div < d_{high}$
            $v_{id} = \omega \times v_{id} + c_1 \times r_1 \times (pbest_{id} - x_{id}) - c_2 \times r_2 \times (gbest_{id} - x_{id})$
        Elseif $Div < d_{low}$
            $v_{id} = -1 \times \omega \times v_{id} - c_1 \times r_1 \times (pbest_{id} - x_{id}) - c_2 \times r_2 \times (gbest_{id} - x_{id})$
        Update new-Position using Eq.1
        Calculate fitness Function using Eq. 3
    End for-loop
End while

The main steps of ATREL-PSO algorithm are described in Algorithm 1. The first of the two new line in Algorithm, initialize all parameters and variable, and then for each robot calculate diversity. The next velocity and position of each robot depend on the value of fitness function as well as diversity value *(Div)*. The diversity of the swarm is measured according to the following formula:

$$diversity(Div) = \frac{1}{n}\sum_{i=1}^{n_s}\sqrt{\sum_{j=1}^{n_x}(x_{ij}(t) - \overline{x_j(t)})^2} \quad (6)$$

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

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

$$\overline{x_j(t)} = \frac{\sum_{i=1}^{n_s} x_{ij}(t)}{n_s} \tag{7}$$

The values of d_{high} and d_{low} that influence the efficiency of ATREL-PSO, express the upper bound and the lower bound of the diversity of species respectively. The higher values for 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 d_{high} and d_{low} should be neither too low nor too high and we can choose from the experiential values. When a robot is close enough to the goal, it can change its search mechanism from ATRE-PSO to the local search according to its fitness function. The reason behind this change is that local search is able to guide the robot toward the goal better when the robot is near the goal, whiles the PSO may guide the robot escape from the goal. In the local search method, each robot in its current location has some adjacent safe positions that the robot can move towards them with specific direction, but first the robot checks which direction has the best fitness function and then moves to the specific adjacent position along the same direction.

4. RESULT AND DISCUSSION

4.1 Simulation Condition

The simulations were performed in Visual Basic 6.0 software and the results of the proposed algorithm ATREL-PSO, ATRE-PSO and basic PSO, on a group of agents (*i.e.*, robots) are presented. The number of particles in the population is equal to the number of robots, so each particle represents a single robot. Robots are randomly deployed in the search space. Since all *ATREL-PSO, ATRE-PSO and Basic PSO* are stochastic algorithms, every time they are executed they may lead to different trajectory convergence. Therefore, multiple test groups of 100 trials of 400 iterations for each algorithm 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 are chosen for the target in four different place of the search space (see Figure 3).

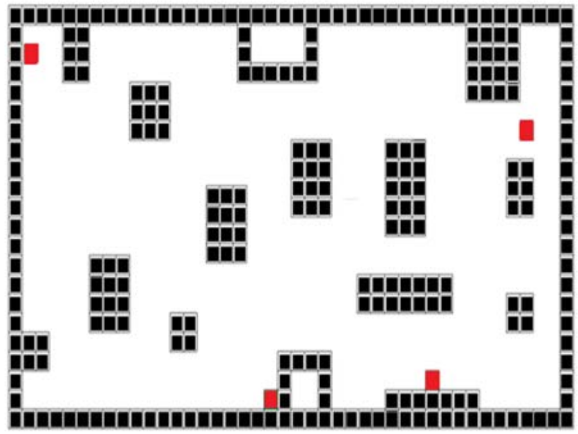


Figure 3. The map of simulation search space and the four different target point locations

4.2 Simulation Result

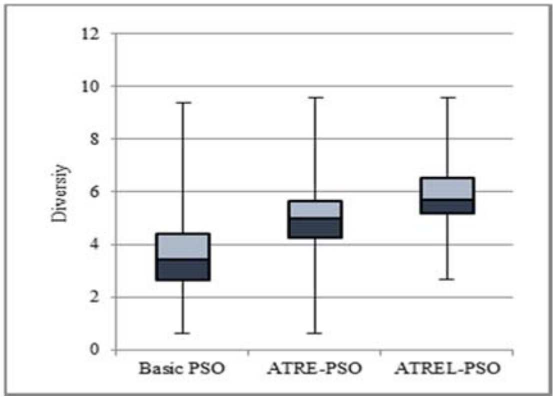
To verify the effectiveness of the proposed algorithm, we present several experiments with respect to the number of robots and types of environment. In section 4.3 we study the performance of the proposed algorithm (ATREL-PSO) in three different environments like without obstacle, with obstacle and complex environment. We also compare it with other searching algorithms (Basic PSO and ATRE-PSO) in the three different environment. Section 4.4 presents the search time of ATREL-PSO and compare with Basic PSO and ATRE-PSO in the worst-case scenarios. In this scenario the search time consumed in averaged 100 runs by ATREL-PSO, ATRE-PSO and Basic PSO in three different environments with an increasing number of obstacles is evaluated. The maximum number of iteration is considered to be 400, which is obtained by trial and error.

4.3 Diversity Evaluation

To evaluate the diversity of the *ATREL-PSO, ATRE-PSO and Basic PSO* we made several simulation runs in three different environments: without obstacle, with obstacle and complex environment. We used the combination of four different target position and robot positions to make the worst case in each test case. In the other words, we try to put the target in the farthest place towards the initial robot positions in each test case that the robots cannot see it easily and need to search the more regions. It should be noted that the diversity of the algorithms was calculated according to the Eq. (6), (7).

4.3.1 Diversity evaluation in environment without obstacle

The search space explored by the robots in the environment without obstacle. Here, figure 4, shows the diversity of ATREL-PSO, Basic PSO and ATRE-PSO in environment without obstacles. In this case, the performance of both ATREL-PSO and ATRE-PSO is similar, and better than Basic PSO. It is clear from the Figure that the diversity of all three algorithms is quite similar. However, the diversity of Basic PSO is lower than the other two algorithms, this is because of the attraction and repulsion property of these two algorithms. In addition the diversity of ATREL-PSO algorithm is slightly higher than ATRE-PSO and this is because in this environment the robots could observe the target easily and change the mechanism to A* to get the target.



Firgure 4: Diversity of ATREL-PSO, ATRE-PSO and Basic PSO in environment without obstacle.

4.3.2 Diversity evaluation in environment with obstacle

The diversity of ATREL-PSO, ATRE-PSO and Basic PSO are compared in the environment with 10 obstacles in 100 test cases. This environment is more complex than the previous environment, and we expect the weakness of Basic PSO to be more evident than before. Figure 5 presents the diversity of these algorithms. In this case the probability of observing target is lower than the previous case. It can be seen that the diversity of Basic PSO is lower than ATRE-PSO and the proposed methods. In this case, the static obstacles could not allow the Basic to get out of the local optima environment and because of its low diversity property it stuck into the local optima and cannot explore the other regions. Therefore, the diversity of this algorithm is lower than the others. However, the diversity of ATRE-PSO and ATREL-PSO is higher in the same situation and this because of the attraction and repulsion feature. It is also shown that the diversity of our proposed algorithm is slightly better than ATRE-PSO (around 8), while the diversity of ATRE-PSO is about 7, and this is because of adding Local search method (A*).

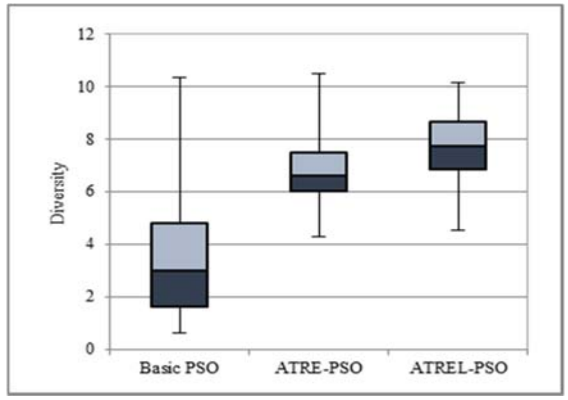


Figure 5. Diversity of ATREL-PSO, ATRE-PSO and Basic PSO in environment with obstacle.

4.3.3 Diversity evaluation in complex environment

This environment is the most complex environment in this study, and contains 14 obstacles. The diversity of all three algorithms are shown in Figure 6. In this situation the diversity of Basic PSO is significantly lower than the other two algorithms and it shows that it cannot reach the target in the complex environment. In this environment Basic PSO algorithm stuck into the local optima and searched only the same environment in the desired time and could not escape from that region. This shows that the Basic PSO could not explore the other areas in case of having more obstacles. As a result, it diversity is low. On the other hand the ATREL-PSO and ATRE-PSO could explore the environment and reach the target in a given search time. The figure shows that the diversity of ATREL-PSO is quite similar to ATRE-PSO.

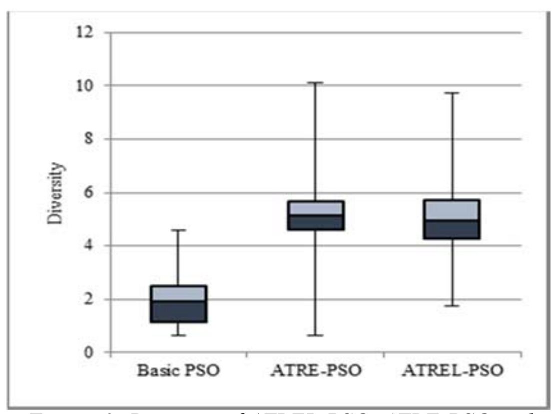


Figure 6. Diversity of ATREL-PSO, ATRE-PSO and Basic PSO in complex environment

It shows that in most of the test cases Basic PSO algorithm stuck into local optima and could not reach the target. In the complex environment, the diversity of ATREL-PSO and ATRE-PSO is two times more than Basic PSO, and the diversity of Basic PSO in this environment is less than the other two environment. It means that in the complex environment Basic PSO algorithm easily stuck into the local optima and could not escape from that regions.

4.4 Overall Performance of The Proposed Algorithm (ATREL-PSO)

In this part the overall performance of ATREL-PSO, ATRE-PSO and Basic PSO are studied and compared with respect to the three different environments (without obstacle, with obstacle and complex environment) as well as four target positions. We run 100 test cases for each environment with four target positions. At each test case the initial position of each robot is randomized. After running 100 test cases, we averaged the success rate and the number of iterations that each algorithm used to reach the target. It should be noted that the simulation stops when the robots reach the target or the maximum number of iterations (400 iterations) has elapsed. The performance of the algorithms are evaluated based on the search time that each algorithm reach the target. If the robot controlled by each algorithm reaches the target in lesser time then that algorithm will have a better performance.

Table 1, compares the average number of iterations and success rate (%) of three algorithms in environment without obstacle. All the experiments in Table 1, was fully successful and all the algorithm could reach the target. However, all the search algorithm could reach the target, it can be seen that the average number of iterations in Basic PSO is far greater than ATREL-PSO and ATRE-PSO. It is also shown that the average number of ATREL-PSO is less than ATRE-PSO which indicate the advantage of using A^* local search to overcome the problem of exploration when the robot is close to the target.

It also shows that when the target is closer to the initial position of robots then the overall performance of Basic PSO is close to the other algorithm. However, when the initial position of the robots in the swarm is far from the target position (position 1) the success rate of Basic PSO is low. It is clear from the Table that both ATREL-PSO and ATRE-PSO are not sensitive to target position and in the worst case scenario, when the target is placed far from the initial position of the target they could successfully reach the target. However, the success rate of ATREL-PSO is better than ATRE-PSO in target position 1. In this case, ATRE-PSO may move toward the target by the attraction property but then move away from the target by repulsion property.

Table 1: Average and success rate for environment without obstacle using different Target position.

Basic-PSO		**ATRE-PSO**		**ATREL-PSO**		**Target position**
Avg	Succ (%)	Avg	Succ (%)	Avg	Succ (%)	
210.15	**75**	**98.25**	**95**	**95**	**100**	**1**
177.64	**92**	**74.47**	**100**	**68.54**	**100**	**2**
95.32	**95**	**72.12**	**100**	**73.78**	**100**	**3**
89.85	**97**	**56.35**	**100**	**45.25**	**100**	**4**

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

Table 2: Average and success rate for environment with obstacle using different Target position.

Basic-PSO		ATRE-PSO		ATREL-PSO		Target position
Avg	Succ (%)	Avg	Succ (%)	Avg	Succ (%)	
275.43	**42**	**118**	**89**	**75**	**100**	**1**
225.25	**45**	**74**	**92**	**55.71**	**100**	**2**
172.87	**56**	**72.12**	**100**	**63.78**	**100**	**3**
165.85	**59**	**56.35**	**100**	**45.25**	**100**	**4**

Table 2 compares the overall performance of three algorithms in the environment with obstacles. It shows that the success rate of Basic PSO in this environment is significantly decreased and in only 50 % cases could achieve the target. It also shows that the average number of iterations for Basic PSO is higher in compare to the previous environment (without obstacle). However, the success rate of ATREL-PSO in all four target position is 100% and it means that the algorithm could guide the robots properly to achieve the target. It should be noted that average number of iteration using this algorithm is low. Therefore, the overall performance of our proposed algorithm is higher than the other two algorithms. The success rate of ATRE-PSO is significantly higher than Basic PSO but lower than ours. It shows that the performance of ATRE-PSO is not as good as ATREL-PSO. The average number of iteration using ATRE-PSO is slightly lower than the ATREL_PSO.

Table 3 compares the average number of iterations and success rate (%) of all three algorithm in complex environment using four target search algorithms. It is shown that the overall performance of Basic PSO in this environment is less than 30% success, which shows the premature convergence problem. In fact, Basic PSO algorithm in this environment could not find the target in most of the cases and stuck into the local optima. In the target position one Basic PSO algorithm could not achieve the target in none of test cases. The success rate of ATRE-PSO is slightly lower than previous environment (with obstacle). This is due the fact as the number of obstacles increase the robots stuck between the obstacles and could not observe the target easily. This algorithm only help the robots to increase the diversity and explore the different region but it cannot guide them to move toward the target when the robot observe the target. For example in target position one, when the target is placed in the farthest place to the robots initial positions, the success rate of the algorithm is lower in compare to the other target positions. The average number of iteration using ATRE-PSO is significantly higher in compare to the previous environment (with obstacles).

In the complex environment (Table 3), in one case (Target position 1), Basic PSO failed and could not achieve the target. In this scenario, the target is places behind the obstacles and the robots could not see it easily. This Target position shows that the Basic PSO is not applicable in the complex environment, however the other two algorithm could reach the target even with high average number of iterations. The success rate of ATREL-PSO is 100% in all scenarios, however, the success rate of ATRE-PSO is not 100% in all scenarios. It is because of high diversity in the ATRE-PSO even when the robots is close to the target. The proposed method (ATREL-PSO), create and efficient balance between exploration and exploitation.

However, the success rate of our proposed algorithm is 100%, and it shows that the performance of this algorithm is not sensitive to the number of obstacles and target position. It should be mentioned that the average number of ATREL-PSO is less than ATRE-PSO and it is because of utilizing A^* algorithm in this method which reduce the number of iteration and help robots get the target faster.

The average number of iterations of the other two algorithms (ATRE-PSO and ATREL-PSO) is decreased dramatically in this environment and it shows the high diversity of these two algorithms.

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

Table 3: Average and success rate for Complex environment using different Target position.

Basic-PSO		ATRE-PSO		ATREL-PSO		Target position
Avg	Succ (%)	Avg	Succ (%)	Avg	Succ (%)	
-	**Fail**	**228**	**78**	**205.67**	**100**	**1**
277.64	**15**	**114**	**86**	**95**	**100**	**2**
195.32	**24**	**92.12**	**95**	**93.78**	**100**	**3**
189.85	**29**	**86.35**	**100**	**74.25**	**100**	**4**

5 CONCLUSION

Maintaining a high diversity while keeping fast convergence are two contradicting features. Multi-robot search algorithm (ATREL-PSO) while maintaining a high level of diversity, decreases the searching time and gave a better performance than the Basic PSO and ATRE-PSO in different initial robot and target position. The features presented in this study were implemented in a simulation environment and experimental results show that the diversity of ATREL-PSO in the multi-robot search systems is better than the Basic PSO in the environment contains static obstacles and robots using this algorithm can find the target faster in a complex environment. Despite the promising result, for the future work it is worthwhile exploring development of other state-of-the-art works that can overcome the premature convergence problem and reduce searching time, and compare with our proposed method.

ACKNOWLEDGEMENT

The authors gratefully acknowledge use of the services and collaboration of Universiti Kebangsaan Malaysia funded by grant GP-K010400 and Vision and Signal Processing Lab at Queensland University of Technology.

REFERENCES

[1] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira, "A Novel Multi-Robot Exploration Approach Based On Particle Swarm Optimization Algorithms," In *Safety, Security, and Rescue Robotics (Ssrr), 2011 Ieee International Symposium On*, 2011, Pp. 327–332.

[2] C. Di Chio, R. Poli, And P. Di Chio, "Extending The Particle Swarm Algorithm To Model Animal Foraging Behaviour," *Lecture Notes In Computer Science*, Vol. 4150, P. 514, 2006.

[3] S. Doctor, G. K. Venayagamoorthy, And V. G. Gudise, "Optimal Pso For Collective Robotic Search Applications," In *Evolutionary Computation, 2004. Cec2004. Congress On*, 2004, Vol. 2, Pp. 1390–1395.

[4] G. Kantor *Et Al.*, "Distributed Search and Rescue with Robot and Sensor Teams," In *Field and Service Robotics*, 2003, Pp. 529–538.

[5] E.-M. Wong, F. Bourgault, And T. Furukawa, "Multi-Vehicle Bayesian Search For Multiple Lost Targets," In *Robotics And Automation, 2005. Icra 2005. Proceedings of the 2005 Ieee International Conference On*, 2005, Pp. 3169–3174.

[6] J. M. Hereford, "A Distributed Particle Swarm Optimization Algorithm for Swarm Robotic Applications," In *Evolutionary Computation, 2006. Cec 2006. Ieee Congress On*, 2006, Pp. 1678–1685.

[7] J. M. Hereford, M. Siebold, and S. Nichols, "Using the Particle Swarm Optimization Algorithm for Robotic Search Applications," In *Swarm Intelligence Symposium, 2007. Sis 2007. Ieee*, 2007, Pp. 53–59.

[8] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," In *Micro Machine And Human Science, 1995. Mhs'95., Proceedings Of The Sixth International Symposium On*, 1995, Pp. 39–43.

[9] S. Xue, J. Zhang, and J. Zeng, "Parallel Asynchronous Control Strategy for Target Search with Swarm Robots," *International Journal of Bio-Inspired Computation*, Vol. 1, No. 3, Pp. 151–163, 2009.

[10] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A Pso-Based Mobile Sensor Network for Odor Source Localization in Dynamic Environment: Theory, Simulation and Measurement," In *Evolutionary Computation,*

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

2006. Cec 2006. Ieee Congress On, 2006, Pp. 1036–1043.

[11] L. Marques, U. Nunes, And A. T. De Almeida, "Particle Swarm-Based Olfactory Guided Search," *Autonomous Robots*, Vol. 20, No. 3, Pp. 277–287, 2006.

[12] J. Pugh, L. Segapelli, and A. Martinoli, "Applying Aspects of Multi-Robot Search To Particle Swarm Optimization," In *Ant Colony Optimization and Swarm Intelligence*, Springer, 2006, Pp. 506–507.

[13] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," In *Swarm Intelligence Symposium, 2007. Sis 2007. Ieee*, 2007, Pp. 332–339.

[14] B. Nakisa And M. N. Rastgoo, "A Survey: Particle Swarm Optimization Based Algorithms To Solve Premature Convergence Problem," *Journal Of Computer Science*, Vol. 10, No. 9, Pp. 1758–1765, 2014.

[15] B. Nakisa, M. N. Rastgoo, M. F. Nasrudin, and M. Z. A. Nazri, "A Multi-Swarm Particle Swarm Optimization with Local Search on Multi-Robot Search System," *Journal of Theoretical and Applied Information Technology*, Vol. 71, No. 1, Pp. 129–136, 2015.

[16] B. Nakisa, M. N. Rastgoo, And M. J. Norodin, "Balancing Exploration And Exploitation In Particle Swarm Optimization On Search Tasking," *Research Journal Of Applied Sciences, Engineering And Technology*, Vol. 8, No. 12, Pp. 1429–1434, 2014.

[17] M. N. Rastgoo, B. Nakisa, and F. S. Najafabadi, "Inspiring Particle Swarm Optimization on Multi-Robot Search System," *International Journal on Computer Science and Engineering*, Vol. 6, No. 10, P. 338, 2014.

[18] M. N. Rastgoo, B. Nakisa, And M. Ahmadi, "A Modified Particle Swarm Optimization On Search Tasking," *Research Journal Of Applied Sciences, Engineering And Technology*, Vol. 9, No. 8, Pp. 594–600, 2015.

[19] M. N. Rastgoo, B. Nakisa, and M. Z. A. Nazri, "A Hybrid of Modified Pso and Local Search on A Multi-Robot Search System," *International Journal of Advanced Robotic Systems*, Vol. 11, 2015.

[20] R. Poli and C. R. Stephens, "Constrained Molecular Dynamics as a Search and Optimization Tool," In *Genetic Programming*, Springer, 2004, Pp. 150–161.

[21] M. S. Couceiro123, F. M. Martins, R. D. J. Iii-Solum, R. P. Rocha, And N. F. Ferreira, "Analysis And Parameter Adjustment Of The Rdpso Towards An Understanding Of Robotic Network Dynamic Partitioning Based On Darwin's Theory," In *International Mathematical Forum*, 2012, Vol. 7, Pp. 1587–1601.

[22] B. Nakisa, M. N. Rastgoo, D. Tjondronegoro, And V. Chandran, "Evolutionary Computation Algorithms For Feature Selection Of Eeg-Based Emotion Recognition Using Mobile Sensors," *Expert Systems With Applications*, 2017.

[23] Y. Shi, "Particle Swarm Optimization: Developments, Applications and Resources," In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress On*, 2001, Vol. 1, Pp. 81–86.

[24] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Ieee Transactions On Systems Science And Cybernetics*, Vol. 4, No. 2, Pp. 100–107, 1968.

[25] F. Liu, A. Narayanan, And Q. Bai, "Effective Methods For Generating Collision Free Paths For Multiple Robots Based On Collision Type," In *Proceedings Of The 11th International Conference On Autonomous Agents And Multiagent Systems-Volume 3*, 2012, Pp. 1459–1460.

[26] M. Pant, T. Radha, and V. P. Singh, "A Simple Diversity Guided Particle Swarm Optimization," In *Evolutionary Computation, 2007. Cec 2007. Ieee Congress On*, 2007, Pp. 3294–3299.

[27] M. N. Rastgoo, B. Nakisa, M. F. Nasrudin, A. Nazri, and M. Zakree, "A Critical Evaluation of Literature on Robot Path Planning In Dynamic Environment." *Journal of Theoretical & Applied Information Technology*, vol. 70, no. 1, 2014.