Inspiring Particle Swarm Optimization on Multi-Robot Search System

[1]Mohammad Naim Rastgoo, [1]Bahareh Nakisa,[2]Farzaneh Salimian Najafabadi

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

[2]Department of Electrical Engineering, Najafabad Branch,Islamic Azad University, Isfahan, Iran

***Abstract*-Multi-robot Search system is one area that attracts many researchers. In the field of multi-robot system one of the problem is to design a system that allow the robot to work within a team to find a target. There are many methods that are used on the multi-robot systems. One of the methods is Particle Swarm Optimization (PSO) that uses a virtual multi-agent search to find a target in a 2 dimensional search space. In this paper we present a multi-search algorithm by modifying the Particle Swarm Optimization algorithm to model an abstracted level theeffects of changing aspects and parameters of the system suchas number of robots.**

Keywords: Particle Swarm Optimization, Multi-robot system, Searching task

I. INTRODUCTION

Robotic search are used in many real-world applications like search and rescue, object finding, hazardous waste cleanup [1] and planet exploration [2]. By growing the technology, researchers felt the need to use multi-robot systems to solve the problems of mobile robotics in a more efficient way and also to tackle the problems that are difficult for single robotic systems to solve. Using Multi-robot in searching task decrease the search time to find the located target and improve the robustness against failure of single robots by redundancy as well as individual simplicity. Since the swarm provide more environmental information and sensory then the decision making in this system is stronger than the single agent. Although search has been well explored in the past [3], using multi-robot systems for search is a more recent development and has not yet been studied extensively.

The time consuming to collect experimental data with multi-robot system in the real world is high therefore this limitation motivates the use of abstracted model which uses approximations of details of the system which have little impact on the targeted performance metrics. There are two categories for such models: Microscopic and Macroscopic. Macroscopic models a swarm robotic as a whole and Microscopic models each robot separately.

In 1995 James Kennedy and Russell Eberhart introduced Particle Swarm Optimization (PSO) that was based on social behavior of fish schooling or bird flocking [4, 5]. PSO is one of the evolutional optimization methods and can solve many optimization problems that are encountered in various fields of technology. This method because of the simple concept and easy implementation has developed fast in recent years .In bird flocking, a flock of birds is looking for food and tries to keep following the members of the group that have the closest position to the food. The birds that have the better situation inform others and they simultaneously move toward that place. By doing this process iteratively and communication among them, they might achieve the better places and find the food faster. Particle Swarm Optimization by imitating from the social behavior of bird flocking, initializes a population of particles that simulates a flock of birds. The particles that each is represented as a solution are spread out in the search space randomly and search for finding the optimal or near optimal solution by generating new solutions. Each particle is represented with its position that is as a set of coordinates, which describes a point in a search space and its velocity and it's best past position achieved so far. At each iteration of the algorithm, particles in their current positions are evaluated through fitness function and if the value of fitness function is better than any that is found so far, it is stored as the best position called P_{best}. The particle with the closest position to the goal gets the highest value in fitness function and is stored as G_{best}. After that, the next position that particle has to go and also its velocity is calculated by the following formula:

$$v_{i+1} = w.v_i + \varphi_p.r_p\left(P_{best_i} - x_i\right) + \varphi_g.r_g\left(G_{best_i} - x_i\right) \tag{1}$$

$$x_{i+1} = v_{i+1} + x_i \tag{2}$$

where w is the inertia coefficient which slows the velocity over time to prevent explosions of the swarm and ensure ultimate convergence,φ_p is the weight given to the attraction to the previous best location of the current particle and $\varphi_g$is the weight given to the attraction to the previous best location of the particle neighborhood. $r_g, r_p$are the random numbers between (0,1) generated at each iteration randomly for each particle.

The parallel between the multi-agent search in the robotic scenario and the multi-agent search in the virtual optimizationspace has been recently explored in several instances.Distributed unsupervised robotic learning was accomplishedin a robotic group by assigning each robot a unique PSOparticle that represented the robot controller [6]. Adaptationsof PSO have been used for multi-robot odor search in severalinstances [7,8]. Particle

Swarm Optimization was alsoapplied recursively to a multi-robot search task, where the parameters of the PSO-inspired search were optimized by anexternal PSO algorithm [9]. The effect of including aspectsof multi-robot search in PSO has been partially explored[10]. Additionally, PSO was used as an inspiration for asolution to a multi-animal foraging task [11], which could beapplied to multi-robot systems as well. However, none of theseapplications extend the inspiration to use PSO as an effectivemodel of the robot group performance.

II. TECHNIQUES

By Modifying the PSO algorithm and applying on the multi-robot search system we can generate an effective search algorithm.In this paper we exchange the particles, Fitness function and Continues Search space in Basic PSO to Robots, Camera of the robots and Discrete Search space.

A. *Discrete Search space*

The real space has transformed into two-dimensional search space that are divided into squares. Each square, which is called a cell, 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 a single target and there are no obstacles in the search space. Each cell in the search space is marked as a safe cell or unsafe cell. If a cell is occupied by a robot or target this cell is marked as an unsafe cell otherwise is marked as a safe cell. To prevent the collision between the robot and other robots, the robot should move to the safe regions. When the robot stands in a cell, it visits the center of that cell and is considered as a visited cell, therefore that cell in that timestep is marked as an unsafe cell. The search space in this study has a boundary and the robots cannot go out the search space. If the next position of the robot is placed out of the search space then by reversing the direction of the robot velocity, the next position of it is placed into the search space.

B. *Particles versus Robots in multi-robot system*

The particles in the PSO are matched with robots in the multi-robot system. In this paper we assumed that each robot by accessing to the map of the search has complete knowledge about its location. The geometrical shape of the robot is assumed to like a circle with the determined radius (☐) 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. 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 ([12]). 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 particle can update its position by using the PSO equitation at every iteration. Then with appropriate velocity in a fixed amount of time move toward the next position. The robots are synchronized in this method so that the iterations match between robots. The Particles in Basic PSO do not have the limited acceleration and velocity but the robots in the real world have limitation on how quickly can move. Therefore, the velocity is discretized into discrete values that enable it executes just one action at each time step. The velocity is limited between $[-V_{max}, V_{max}]$ where the v_{max} represent the maximum velocity of the robot along its direction and the $-v_{max}$ is the maximum velocity of the robot but in the reverse direction. When the velocity of the robot reach higher than the maximum velocity value then it is assigned the maximum velocity value.

Another important factor, which has to be considered for adopting Basic PSO algorithm on multi-robot search system, is about the collision between robots. In multi-robot system, robots and the target have some volume therefore they have to prevent to collide with each other or static obstacles. Using the standard PSO particle displacement at each iteration, we will be unable to detect any collisions that might occur along the path. To detect the collision between robots we need to divide the continuous movement of them into multiple steps and check their movement at each step. In order to prevent from the possible collision between robots in the search space we use the method that is introduced by ([13]). 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.

C. *Robot Camera as a Fitness function*

In this paper it is assumed that each robot is equipped with one camera that capture the picture from it's surrounding. At every iteration, each robot applies its camera to calculate its fitness function. If the desired target is placed into range of view of the camera then the fitness function is calculated based on the equation (3) otherwise it returns Zero.

$$0 < fitness\ function = \frac{\sum_{i=1}^{n} PT_i}{\sum_{j=1}^{m} PC_j} < 1 \tag{3}$$

$PT = \{PT_1, PT_2, PT_3 \dots PT_n\}$is a set of pixels of the target in the image captured by the camera and $PC = \{PC_1, PC_2, PC_3 \dots PC_n\}$ is a set of pixels in the image captured by the camera. The camera of the robot can rotate in 8 different directions.Therefore, it has the ability to observe the entire environment by rotating it 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 cells.

III. ANALYSIS OF PSO-INSPIRED SEARCH WITH GLOBAL POSITIONING

We now simulate our PSO-inspired algorithm on multi-robot search system and compare its performance with Basic PSO algorithm on single robot. The performance of the PSO algorithm in different situation is compared when the number of robots increases. Search time is applied as a measurement to compare the performance of the algorithm with different number of robots. When a robot finds the target and reaches it in lesser search time in an experiment it will have a better performance.

A. Setup

In our simulation, the robots are initially spread in the search space randomly with random velocity and headings. The Modified*PSO* is simulated and tested in five different initial robots position and a target position. The target is placed in five random position of the search space. The Figure 1 shows the search space and the target in one of the five different positions.In this study the search time was selected as a measurement to compare the performance of algorithms. In each initial robots position, each algorithm performed 400 test cases.To calculate the next velocity, there are two random values (r_1, r_2) that are randomly selected in each test case.

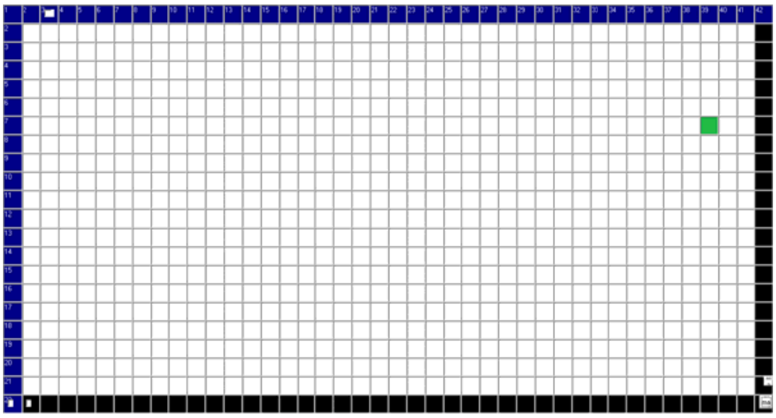

Figure 1. Map of simulation search space and the target point locations.

The overall performance of Modified *PSO* was evaluated (i.e., the number of iterations that were found) and then compared in different number of robots. Figure 2 compares the performance of *Modified PSO* in five different initial robots positions. The figures show the search times (number of iterations passed) for*Modified PSO* algorithm. In this Study, the search space is assumed to be bounded with borders and thus, the robots cannot go outside the search space. Due to the condition approximation of the actual robot searching, the search space in our simulation is a hard border. It is assumed that when the next positions of the robots were set out of the search space, it should reverse and be placed inside the search space. For the simulation results the inertia coefficient, □, was set to 0.9…. 0.5 and the both coefficient c_1, c_2 were set to 2. We set an initial value for *v* (velocity) for each robot to simulate the behavior of a physical robot. The Maximum number of robot for this study is three; therefore, the *lbest* topology was identical to the *gbest* topology. In this study, we adapted the PSO algorithm to the multi-robot search system; therefore, unlike most of the PSO studies that have tracked the function value, our simulation searched the target function. The simulation stopped when the robot reach the target or when the maximum number of iterations (200 iterations) occurred.

B. Simulation result

To evaluate the effectiveness of the *Modified PSO,* we look at the time consumed to find the target by the different number of the robots. We used the combination of an initial target positions and five initial robot positions to made the worst case in each test case. In each test case, each algorithm performed 400 runs. The averaged performance of *Modified PSO* was evaluated (i.e., the number of iterations that were found). Figure 2 compares the performance of *Modified PSO* in five different initial robots position and shows the search times (number of iterations passed) for*Modified PSO*.

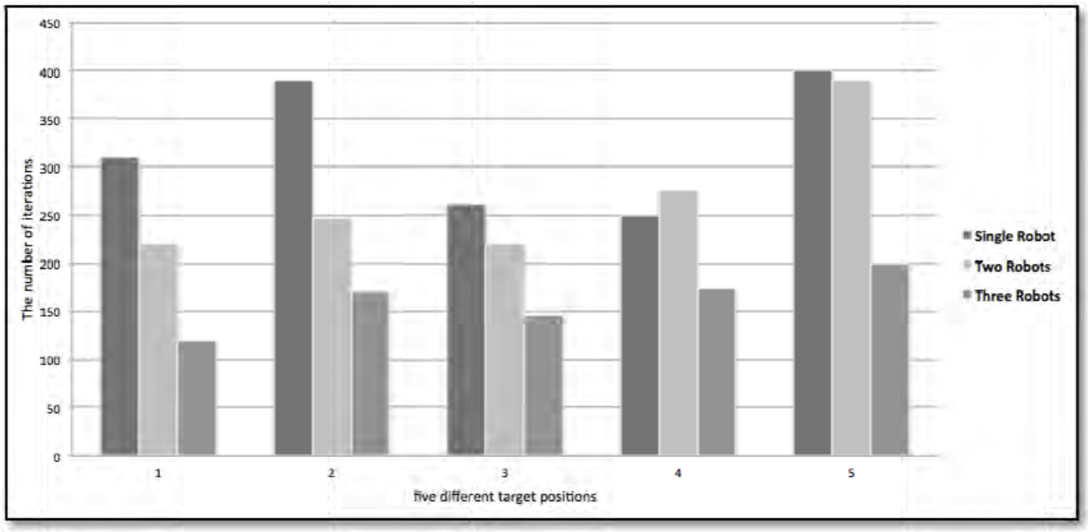

Figure 2: Performance of the PSO algorithm in different target position while the number of robots increases.

As expected, the performance of the proposed algorithms increases when the number of particles/robots increases, as the swarm is better able to explore the environment. We can see the performance of both algorithms with three particles/robots in different target positions outperforms two robots and single robot in Fig. 2.As described before, it isassumed if the robot or robots could not reach the target until 400 numbers of iterations the programis terminated. Therefore, in some cases (Target No.2 and No.5) the robot or robots could not find that is due to behavior of Basic PSO.In other words, In the early stage of Basic PSO algorithm robots explore the search many different search regions but as the time progress the moving diversity of robots decrease and they tend to move toward the same regions and this low diversity leads to stagnation in the fitness of the swarm and they trap into the local optima. In addition, in some cases the robots cannot exploit good promising regions, which will increase the search time. In other words, sometimes the target is very close to the robot (particle) and the fitness value, represent the amount of size of the target that the robot has observed by its camera, is high but the robot must move according to PSO formula and this may guide the robot to move to the position that is placed in farther pose toward the target.

IV. CONCLUSION

We have presented a multi-robot search algorithm based on the Particle Swarm Optimization and shown that the performance of the swarm to find the target is better than the single robot. We have adopted PSO to model this algorithm and achieved close matching between the two. The result shows that Basic PSO on multi-robot search system has some problems, which increase the search time and need to be modified to improve the performance of the algorithm and decrease the search time. One of the major problem of Basic PSO on multi-robot search system is that in the early stage of the search the exploration of the robots are more and as the number of iterations increase the moving diversity of the robots decrease therefore they converge to the same regions and cannot search the different regions in the search space.

V. REFERENCE

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

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

[3] Benkoski, S. J., Monticino, M. G., Weisinger, J. R., 1991. “A Survey of the Search Theory Literature”, Naval Research Logistics, Vol. 38, Nr. 4, pp. 469-494.

[4] Eberhart, R. & Kennedy, J., 1995. “A new optimizer using particle swarm theory” Micro Machine and Human Science, Proceedings of the Sixth International Symposium on, 4-6 Oct, pp. 39-43.

[5] Kennedy, J. & Eberhart, R., 1995. “Particle swarm optimization” Neural Networks, Proc. of IEEE International Conference on, Nov/Dec, pp. 1942-1948.

[6] Pugh, J. & Martinoli, A., 2006.“Multi-Robot Learning with Particle Swarm Optimization”, International Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan, May 8-12, pp. 441-448.

[7] Jatmiko, W., Sekiyama, K., & Fukuda, T., 2006. “A PSO-based Mobile Sensor Network for Odor Source Localization in Dynamic Environment: Theory, Simulation and Measurement”, IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, July 16-21, pp. 1036-1043.

[8] Marques, L., Nunes, U., & de Almeida, A. T., 2006."Particle swarmbased olfactory guided search", Autonomous Robotics, Vol. 20, pp. 277- 287.

[9] Doctor, S., Venayagamoorthy, G. K., & Gudise, V. G., 2004. "Optimal PSO for Collective Robotic Search Applications", IEEE Congress on Evolutionary Computation, June 19-23, Portland, OR, pp. 1390-1395.

[10] Pugh, J., Segapelli, L., & Martinoli, A., 2006. "Applying Aspects of Multi-Robot Search to Particle Swarm Optimization", International Workshop on Ant Colony Optimization and Swarm Intelligence, Brussels, Belgium, September 4-7, pp. 506-507.

[11] Di Chio, C., Poli, R., & Di Chio, P., 2006. "Extending the Particle Swarm Algorithm to Model Animal Foraging Behaviour", International Workshop on Ant Colony Optimization and Swarm Intelligence, Brussels,Belgium, September 4-7, pp. 514-515.

[12] 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.

[13] Liu, F., Narayanan, A., & Bai, Q. 2012. Effective methods for generating collision free paths for multiple robots based on collision type. Paper presented at the Proceedings of the 11th International Conference on Autonomous Agents and Multi-agent Systems-Volume 3.