VNU Journal of Science: Comp. Science & Com. Eng., Vol. 32, No. 3 (2016) 1-10

Swarm Optimization Approach for Light Source Detection by Multi-robot System[1]

Hoang Anh Quy, Pham Minh Trien

VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam

Abstract

Exploration and searching in unknown or hazardous environments using multi-robot systems (MRS) is among the principal topics in robotics. There have been numerous works on searching and detection of odor, fire or pollution sources. In this paper, a modified Particle Swarm Optimization Algorithm (PSO) was presented for MRS on detecting light sources, namely APSO. In the proposed algorithm, an integration of conventional PSO and Artificial Potential Field (APF) is employed to use swarm intelligence for space exploration and light source detection. The formulas for APSO velocities are based on those of PSO and APF. Furthermore, each particle is surrounded by an APF that forms repulsive force to prevent collision while the swarm is in operation. The simulation results of APSO in Matlab by various scenarios confirmed the reliability and efficiency of the proposed algorithm.

Received 04 December 2015, Revised 09 January 2016, Accepted 26 September 2016

Keywords: PSO, MRS, APF, APSO, light source detection.

1. Introduction

Owing to their robustness to local optima, widespread coverage and high degree of accuracy, multi-robot systems (MRS) are highly efficient in the tasks of space exploration and searching in unknown environments. There have been numerous works in which MRS was used to detect fire, pollutant sources and odor sources [1, 2, 3].

Among a variety of potential algorithms to implement on MRS, Particle Swarm Optimization (PSO) has become a natural choice for MRS in searching tasks. PSO was first introduced by Russel Ebenhart and James Kennedy in 1995 [4] and has gained popularity among bio-inspired heuristic algorithms because of its efficiency, intuitiveness and simplicity. Motivated by social searching behavior of natural swarm, PSO is especially effective in optimization problems and widely applied in various fields. Searching tasks of MRS are in fact optimization problems, in which the robots attempt to locate the regions or spots of extreme signal intensity.

Although the idea of applying PSO to multi-robot search is not novel, many problems still need to be addressed adequately in order to put that idea into practice. Some of them are proneness to collision and premature convergence. Many of the related works are concerned with improving performance of the MRS. In [5], the authors concentrated on adjusting learning parameters for better results. In [6] the PSO algorithm was applied to model multi-robot search and the effects of system parameters were also evaluated. In [7], Doctor

[1] This work is dedicated to the 20th Anniversary of the IT Faculty of VNU-UET

[*] Corresponding author. E-mail.: quyha@vnu.edu.vn

et al. proposed a two PSO loops model to control their robot system. The inner loop was applied for collective robotic search and the outer was used to optimize quality parameters of the inner. In [8], Cai et. al. proposed a potential field-based PSO algorithm for cooperative multi-robots in target searching tasks. The problem of premature convergence, which may adversely affect performance of PSO, was addressed in [9], where Nakisa et. al. applied a method based on PSO and Local Search. In spite of various works on application of PSO for MRS in the tasks of exploration or searching in unknown environments, there has not been a standard approach with optimal result. All of the PSO-based algorithms still need further experiments and improvements.

In this paper, we present another approach and a specific application: detecting light sources or in other words, searching for the brightest region in a search space. This method is then compared with one of those mentioned above. In our simulations, an MRS is successfully used to detect light sources (by gathering all the swarm robots around the area of highest luminance in the search space). In all scenarios, each robot (or particle as described in PSO) has to move towards the mutual target and meanwhile avoid obstacles. For the robot swarm to exhibit this behavior, we modified PSO algorithm by associating each particle with an artificial potential field (APF) that can exert repulsive forces to any other particle if their distance is less than a predetermined value called repulsive radius. This method of avoiding collisions is inspired by APF algorithm, which was proposed by Oussama Khatib in 1986 for single robot path planning [10]. APF is widely used nowadays in works on MRS that demonstrate the interaction between robots and obstacles in their work space [11]. The proposed PSO algorithm is named APSO, its details will be presented in the next sections. The simulation in Matlab shows reliable and promising results, which could be applied in various further applications such as dynamic deployment of robotic systems, flame detection or optical wireless charging.

The methodology and simulation are discussed in detail in part 2, the results and discussions follow in part 3. Finally, part 4 concludes this paper with main conclusions and directions for further research.

2. Methodology and simulation

2.1. Methodology

2.1.1. Artificial Potential Field

The APF model is inspired by Artificial Physics with quadratic function, where the choice of coefficients is commensurate to the wireless sensor network of MRS. Myriads of architectures for APF have been developed in accordance with users' definitions and specific tasks, e.g. deploying mobile sensor networks in unknown environment [12], controlling and coordinating a group of robots for cooperative manipulation tasks [13] or maintaining connectivity of mobile networks [14]. In any architecture, magnitude of the potential force existing around each robot is continuously updated based on information collected from its immediate surrounding environment and other robots via connection network. Therefore, APF is used to regulate the relation between robots in term of position. Potential force is categorized into two main groups: passive force and active force. Passive force is generated when robot emit signal and determine distance to neighboring robots or obstacles by the magnitude of reflected signal to avoid obstacle or remain relative position with other robots. The signal used in the application could be infrared, ultrasound, laser or camera [15]. On the contrary, active force is generated from external signals. These signals are usually emitted by other robots and transmitted via communication system [11]. In this research, APF is only utilized for the purpose of collision avoidance and only generates repulsive forces on other particles within repulsive region, as defined in this formula:

$$\mathbf{F}_{APFij} = \left(\mathbf{F}_{max} - kr_{ij}^{2}\right)\frac{\mathbf{r}_{ij}}{r_{ij}} \qquad (1)$$

where F_{max} and k are predetermined constants to regulate the magnitude of potential force, $\boldsymbol{F}_{APFij}$ is the APF force exerted on robot i by robot j. $\mathbf{r}_{ij}$ is the end-to-end distance vector from robot j to robot i. r_{ij} is the module of $\mathbf{r}_{ij}$.

Total force exerted on i-th robot of the system is:

$$\mathbf{F}_{APFi} = \sum_{j=1}^{N} \mathbf{F}_{APFij} \qquad (2)$$

where N is the number of robots, $\boldsymbol{F}_{APFij}$ is zero if i = j. The impact of $\boldsymbol{F}_{APFi}$ on overall velocity is controlled by F_{max} and k. As F_{max} increases, the particle is less likely to approach obstacles. In subsection 2.1.2, this will be discussed further.

2.1.2 APSO for MRS

The main contribution of this paper is to propose and evaluate the efficiency of APSO, a modified PSO algorithm. In this subsection, we briefly present principles of PSO and then explain APSO in detail.

In PSO, the swarm consists of homogeneous particles that can explore the search space collectively. During the exploration, the movement of a particle is controlled by a velocity comprised of three components: inertial, cognitive and social velocity. Cognitive velocity leads the particle towards its personal best position and social velocity leads the particle towards the global best. Inertial velocity guides each particle towards their previous directions and thus keeps particles' movement smooth [16]. Besides, high inertial velocity and cognitive velocity at initial steps make the swarm discover search space better. The social learning factor should be increased and cognitive factor should be decreased throughout the exploration in order to enlarge the swarm's coverage at initial steps and make it converge faster at final steps. The searching process using PSO is implemented in four stages: initializing, updating best positions, updating velocity and position, and finally, checking for stopping criteria. PSO velocities and particles' positions are updated with the following formulae:

$$\mathbf{v}_{inertial} = w \times \mathbf{v}_{t-1} \qquad (3)$$

$$\mathbf{v}_{cognitive} = a_1 \times u_1 \times \varphi\,(\mathbf{p}_{t-1} - \mathbf{x}_{t-1}) \qquad (4)$$

$$\mathbf{v}_{social} = a_2 \times u_2 \times \varphi\,(\mathbf{g}_{t-1} - \mathbf{x}_{t-1}) \qquad (5)$$

$$\mathbf{v}_t = \mathbf{v}_{inertial} + \mathbf{v}_{cognitive} + \mathbf{v}_{social} \qquad (6)$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t \qquad (7)$$

where:

$\mathbf{v}_t$: velocity of the swarm at t (time)

w: inertial factor

a_1 : cognitive coefficient

a_2 : social coefficient

u_1 : random number in [0, 1]

u_2 : random number in [0, 1]

$\mathbf{p}_t$: personal best positions at t

$\mathbf{g}_t$: global best positions at t

$\mathbf{x}_t$: position of the swarm at t

φ(**x**): a matrix function that returns a row vector with each element being Euclidean norm of corresponding column in the matrix argument.

In (4), $\varphi(\mathbf{p}_{t-1} - \mathbf{x}_{t-1})$ returns a vector. Each element of this vector is distance from a corresponding particle to its own best position. It is noteworthy that both position and velocity are vectors, so in the step of updating position, they are added directly to get new position, without any dimensional conflict.

To apply PSO to an MRS, each robot is modelled as a particle of the swarm and their movements in the search space resemble those of ideal particles described above. Actual implementation of PSO for MRS involves additional techniques to solve problems which are not covered in its conventional version, such as collision avoidance. APSO is developed to solve that problem. The steps in APSO are basically the same as those of PSO, but the velocities and positions are updated with APF-based formulae. Artificial potential fields are also created around every particle in the search space. The repulsive force between a particle and another particle or an obstacle is given by:

$$\mathbf{F} = (\mathbf{F}_{\max} - k \times r^2) \times (H(0) - H(r_1)) \quad (8)$$

where *r* is the distance between the two objects, $\mathbf{F}_{\max}$ is the maximum value of the repulsive force. H(x) is Heaviside step function. r_1 is the radius of separation, i.e., repulsive forces are only applicable to particles or points whose distance to each other is smaller than r_1. A robot has a limited sensing range, this range must be larger than r_1. *k* is a parameter dependent upon r_1, it is calculated so that **F** is equal to zero when $r = r_1$. Total repulsive force exerted on a robot is the sum of all the repulsive forces exerted by other objects, according to (8).

$\mathbf{v}_{separation}$ is defined as the forth component velocity, responsible for assuring a collision-free exploration of the MRS. In implementation, $\mathbf{v}_{separation}$ corresponds to total repulsive forces on robots in the swarm.

The set of formulae used to update velocity and position in APSO is:

$$\mathbf{w} = sig(\mathbf{d} \times k + l) \qquad (9)$$

$$\mathbf{v}_{inertial} = \mathbf{w}.\times \mathbf{v}_{t-1} \qquad (10)$$

$$\mathbf{v}_{cognitive} = C \times sig(j\,(\mathbf{p}_{t-1} - \mathbf{x}_{t-1}) \times u + v) \; (11)$$

$$\mathbf{v}_{social} = S \times sig(j\,(\mathbf{g}_{t-1} - \mathbf{x}_{t-1}) \times u + v) \; (12)$$

$$\mathbf{v}_t = \mathbf{v}_{inertial} + \mathbf{v}_{cognitive} + \mathbf{v}_{social} + \mathbf{v}_{separation} \; (13)$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t \qquad (14)$$

where:

d: represents immediate population density at the position of a robot.

. ×: element-wise matrix multiplication

sig(**x**): element-wise sigmoid function on matrix:

$$sig(x)(i,j) = \frac{1}{1 + e^{-x(i,j)}} \qquad (15)$$

k, l, u, v: adjusting parameters used to adjust values of quantities of interest.

C: maximum value of $\mathbf{v}_{cognitive}$

S: maximum value of $\mathbf{v}_{social}$

In Figure 1, the implementation of APSO is presented.

In APSO, sigmoid function is widely used because of an appropriate property of the sigmoid curve. It exhibits a relationship between two quantities, in which the first quantity progresses from a small beginning, then accelerates and approach its climax as the second quantity increase. There are three regions on the curve: beginning, acceleration and saturation region.

```
Algorithm: APSO
1.    Initializing
- Generate the population
- Evaluate objective function
2.    Update personal best position
- For each particle, compare fitness of past positions and choose
the optimum position as its new personal best position
3.    Update global best position
- Compare personal best positions of particles and choose the
optimum position as global best position
4.    Update and regulate velocity
- Update velocity using (13)
- Limit velocity if needed
5.    Update position
- Calculate new position using (14)
- Evaluate objective function for each particle
6.    Check stopping criteria
- Stop if maximum step is reached or the swarm has converged
- Otherwise, come back to step 2.
```

Figure 1. Implementation of APSO.

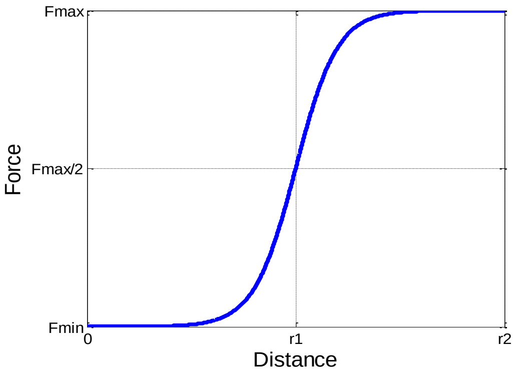


Figure 2. Attractive force.

This property was used to control velocities in APSO. $\mathbf{v}_{cognitive}$ and $\mathbf{v}_{social}$ are dependent upon the distances of particle to their personal best position and global best position. These velocities are regulated so that their magnitude and the corresponding distance could be described by a monotonically increasing relationship. With *k* being negative, (9) gives a lower inertia for a higher population density. During the exploration, each robot sees the search space as a potential field, with repulsive force being proportional to $\mathbf{v}_{separation}$; attractive force proportional to a combination of $\mathbf{v}_{cognitive}$ and $\mathbf{v}_{social}$. The magnitude of attractive force could be described by a sigmoid curve (Figure 2.). The potential field is time-varying. As the position of particles change, global and personal best positions are always improved. Figure 3 shows the potential energy configuration for a robot outside of sensing range of any others. The robot is also not close to any obstacles and its personal best and global best positions are respectively (15, 15) and (20, -20). The search space is confined in x = [-50,50] and y = [-50,50].

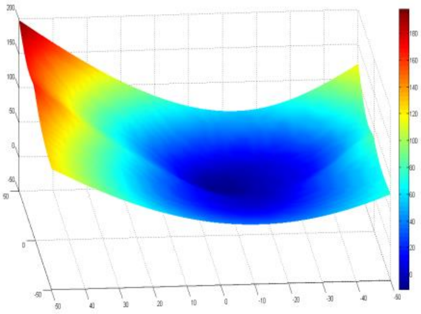


Figure 3. Potential energy configuration.

The main difference between PSO and APSO is how velocity is updated. In APSO, $\mathbf{v}_{separation}$, a new velocity is introduced. Its inertial value depends on immediate population density, $\mathbf{v}_{cognitive}$ and $\mathbf{v}_{social}$ are functions of distance, described by the sigmoid function. This reduces the possibility of collision, meanwhile yields a high performance.

2.1.3. Criteria for convergence

We claim that the exploration is success when the swarm converges atthe point of maximum illuminance. The criterion for convergence of the swarm in conventional PSO is simple and intuitive, as the swarm is said to be converged when all the particles is within a given radius, e.g. 10^{-3} of smallest dimension of the search space, regardless of population size. However, in APSO, such criterion is not applicable because each particle has to maintain a distance to other particles. In our simulation, the two following criteria are used to determine whether the swarm is converged:

1. Improvement in best fitness: The swarm is said to be making progress if in 10 consecutive iterations, best fitness is improved by at least 0.1%.

2. Physical convergence: If in 10 consecutive iterations, the position of the swarm's center of mass does not change considerably (less than the radius of a particle) and a certain number of particles are at a small distance from the center, we said that the swarm has physically converged. The number of particles and the distance are proportional to swarm population.

In short, if there is no improvement in best fitness and the change in the swarm's position is inconsiderable, the swarm is considered to be converged and the searching process is terminated. It is worth noting that this kind of convergence criterion is not absolute convergence since not all particles gather around the swarm's center. The operation is deemed successful if after convergence, the point of highest luminance is covered by the swarm and is within a predefined radius from global best position.

2.2. Simulation

2.2.1. Simulation setup and MRS configuration

In this research, we implement APSO on a homogeneous MRS in Matlab environment. The radius of each robot (r) is set as unit of length. The system has direct communication, the communication range is unlimited (beyond the limit of search space). r_1 is 5×r, i.e. a robot can detect obstacles at the distance of 5×r from its position. Population size varies between 5, 10 and 15. Maximum velocity is 1.5×r/step. Each robot is able to acquire the illuminance at its position via a light sensor on top.

If we set r = 1, the search space size is 100×100. In the Cartesian coordinate system, the ranges of x and y coordinates are both [-50, 50]. We evaluate the effectiveness of the modified PSO algorithm in three scenarios with the presence of an isotropic source and two real light sources: 87517M56FG [17] and AVL1XMAMDG [18]. In the simulations, all obstacles in the search space are static cylindrical obstacles. The radii of cylindrical obstacles used in all scenarios are 4.

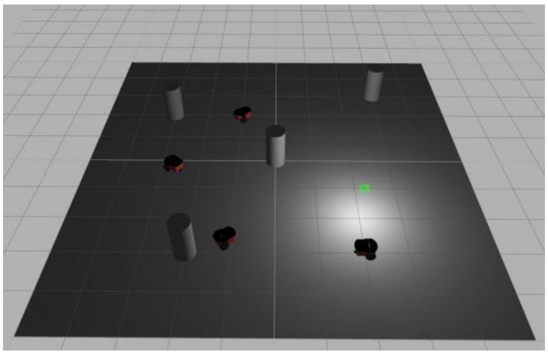


Figure 4. Scenario 1 - 3D View.

2.2.2. Detection of light sources in different scenarios

In the first scenario, a single light source is placed above the search space at (20, -20) (Figure 5). There are four static obstacles at (-30, -30), (-20, 30), (0, 0) and (30, 20) as illustrated in Figure 4.

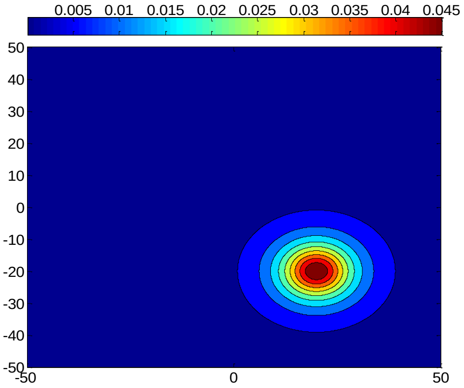


Figure 5. Scenario 1 - Single isotropic light source.

In the next two scenarios, we test with real light sources. Figure 6. and Figure 7. are contour maps of light intensity in regions illuminated by AVL1XMAMDG and 87517M56FG, respectively.

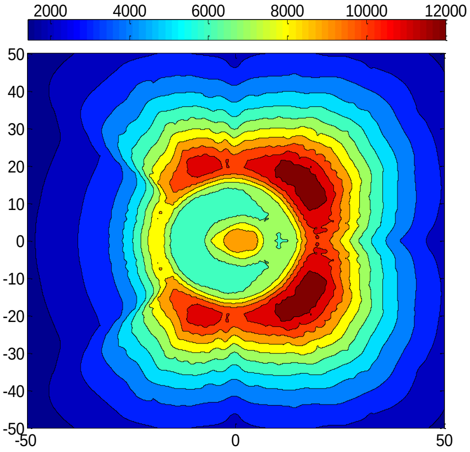


Figure 6. Scenario 2 - AVL1XMAMDG.

In each scenario, three population sizes: 5 robots, 10 robots and 15 robots are simulated. The results acquired after 1000 runs (for each scenario and population size) is presented in Figure 9. The figures are statistical graphs given for analysis of reliability and effectiveness of APSO.

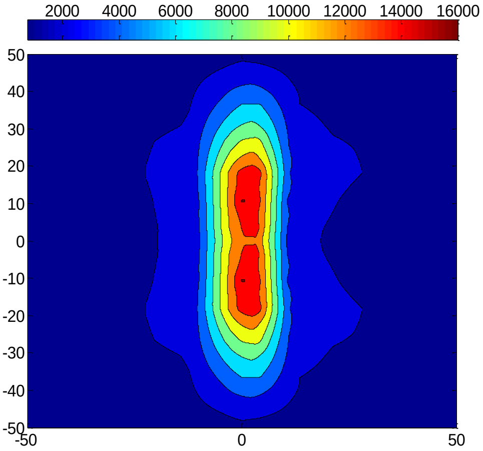


Figure 7. Scenario 3 - 87517M56FG.

In a typical run, as the exploration of this swarm progresses, the robots move towards global best position. As the population size increases and the robots have to maintain a minimum distance from each other, the swarm covers a large area even after convergence. This can be seen clearly in Figure 8.

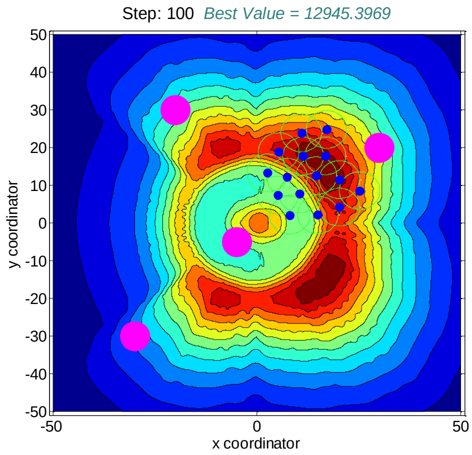


Figure 8. Final distribution of robot swarm - scenario 2.

3. Results and discussion

The main results of these simulations are summarized in the following figures and tables. The results with MPSO - an algorithm from our previous work [19] - are also presented for comparison. Figure 9-11 display the distribution of step of convergence (SC) in each scenario after 100 runs. Figure 12-14 show the cumulative distribution of SC. Only data from successful operations is included.

From the figures, it can be concluded that as the swarm population increases, the step of convergence tends to decrease. However, while there is a large gap in performance between the 5-robot and the 10-robot swarm, there is not much improvement when the population increases from 10 to 15, regardless which algorithm is used. The same pattern can be observed in every scenario.

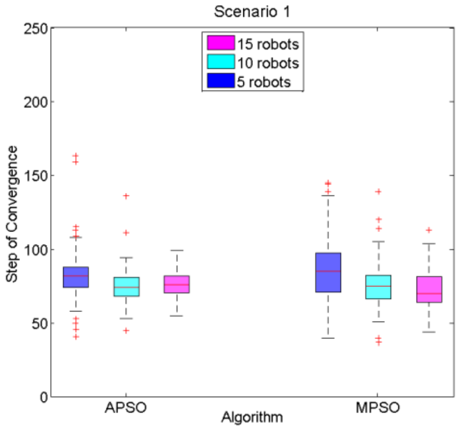


Figure 9. Distribution of SC in scenario 1.

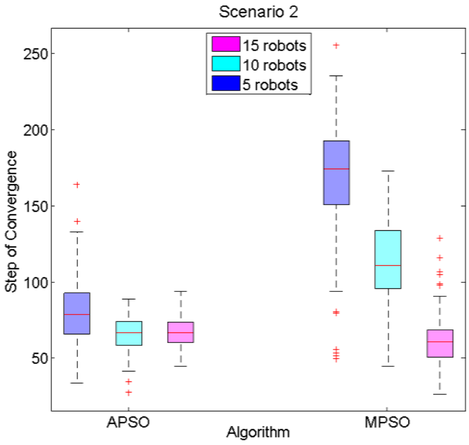


Figure 10. Distribution of SC in scenario 2.

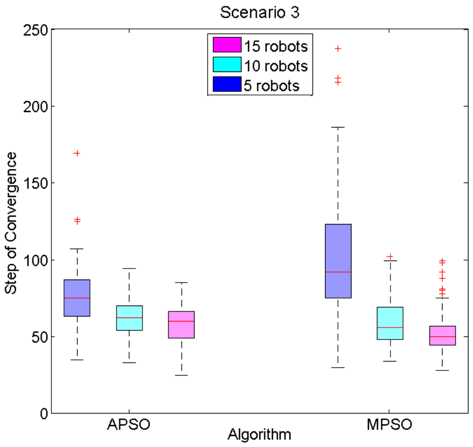


Figure 11. Distribution of SC in scenario 3.

When APSO is applied, there are typically less outliers and IQRs are smaller than when MPSO is applied. We can come to the conclusion that APSO is more stable.

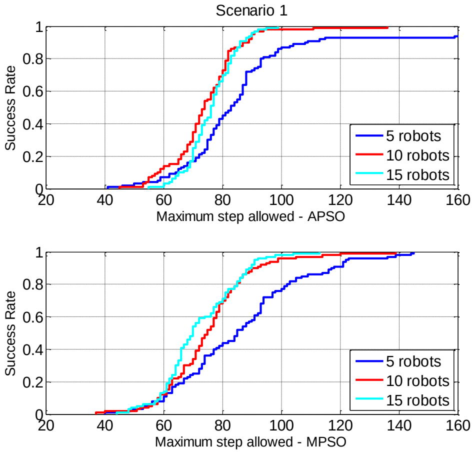


Figure 12. CDF of SC in scenario 1.

Figure 12-14 provide the most accurate way to evaluate the effectiveness of APSO when time (or number of iterations) is limited. In general, to achieve the same rate of success, APSO requires less iterations than MPSO.

In any scenario, if the maximum iteration is 100, success rate of APSO approaches 100% when the swarm population is 10 or 15. The corresponding values of MPSO are all lower. If the maximum iteration is less than 50, there is little possibility that the swarm could converge, no matter which algorithm is chosen. In cases where number of iterations is restricted, due to constraints on energy consumption or time, success rate at a given maximum iteration may become a crucial value to evaluate an algorithm. Table 1 provides data regarding this value, with the maximum iteration being 100.

The data in all the figures consistently indicates low performance of the 5-robot swarm. Both algorithms are not effective for swarms of small population. The swarm with larger initial coverage is less prone to premature convergence.

APSO is also compared to the multi-search algorithm inspired by PSO in the work of Pugh et. al. [6]. With the same constraints and conditions on the robot system, the respective results are given in Figure 15. Initially, the robots are deployed randomly in a square of the size 8×8. The target is placed in the center of the square. The realistic conditions here are wheel slip (10%) and noise (standard normal distribution). In such conditions, APSO even yields better results. In every case, the result is improved when applying APSO.

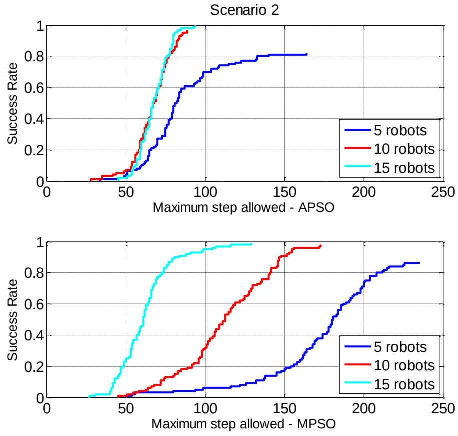


Figure 13. CDF of SC in scenario 2.

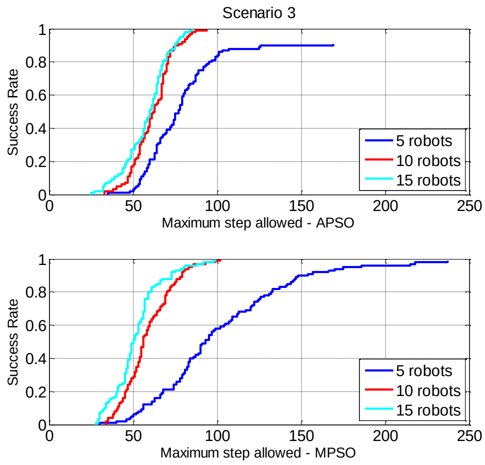
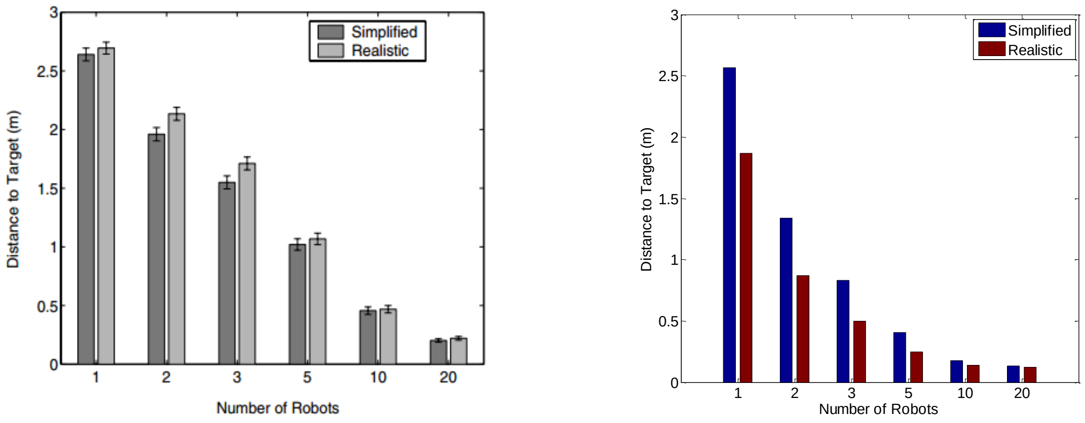


Figure 14. CDF of SC in scenario 3.

Table 1. Success rate at 100th iteration

N	Scenario 1		Scenario 2		Scenario 3	
	A	M	A	M	A	M
5	87	77	70	6	84	58
10	98	96	97	32	100	99
15	100	98	100	95	100	100



a)

b)

Figure 15. Distance to target from the swarm's point of strongest signal detection, averaged over 1000 runs a) multi-search algorithm inspired by PSO. b) APSO.

4. Conclusion and Future works

In this paper, a modified PSO algorithm, namely APSO, is presented for detecting light sources. In this algorithm, APF is integrated into PSO and a new velocity component is introduced to keep the movement of the swarm collision-free. Experimental results in Matlab environment have shown good performance, compared to previous works. With a high success rate, this proposed algorithm is promising for some practical problems involving the utilization of MRS, such as dynamic deployment of robotic systems, flame detection or optical wireless charging.

However, there are still some drawbacks in this algorithm, for example, the swarm is unable to detect multiple sources. Furthermore, it has yet to be tested in complex scenarios.

In future works, we will focus on dealing with them and applying the algorithm on a real MRS.

Acknowledgements

This work has been supported by Vietnam National University, Hanoi, under Project No. QG.15.25. This work is dedicated to the 20th Anniversary of the IT Faculty of VNU UET.

References

[1] A. Marjovi, J. G. Nunes, L. Marques, and A. De Almeida. "Multi-robot exploration and fire searching." In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pp. 1929-1934. IEEE, 2009.

[2] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata. "Plume-tracking robots: A new application of chemical sensors." The Biological Bulletin 200, no. 2 (2001): 222-226.

[3] L. Marques, and A. T. de Almeida. "Finding odours across large search spaces: A particle swarm-based approach." In Climbing and Walking Robots, pp. 419-426. Springer Berlin Heidelberg, 2005.

[4] J. Kennedy, and R. Eberhart, “Particle Swarm Optimization”, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV, pp. 1942-1948, 1995.

[5] Y. Tang, Q. Li, L. Wang, C. Zhang, and Y. Yin. "An improved PSO for path planning of mobile robots and its parameters discussion." InIntelligent Control and Information Processing (ICICIP), 2010 International Conference on, pp. 34-38. IEEE, 2010.

[6] J. Pugh, and A. Martinoli. "Inspiring and modeling multi-robot search with particle swarm optimization." In Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, pp. 332-339. IEEE, 2007.

[7] S. Doctor, G. K. Venayagamoorthy, and V. G. Gudise. "Optimal PSO for collective robotic search applications." In Evolutionary Computation, 2004. CEC2004. Congress on, vol. 2, pp. 1390-1395. IEEE, 2004.

[8] Y. Cai, and S. X. Yang. "An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments." International Journal of Control 86, no. 10 (2013): 1720-1732.

[9] 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 71, no. 1 (2015): 129-136.

[10] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots." The international journal of robotics research 5, no. 1 (1986): 90-98.

[11] T. D. Ngo. "LinkMind: Link optimization in swarming mobile sensor networks." Sensors 11, no. 8 (2011): 8180-8202.

[12] A. Howard, M. J. Matarić, and G. S. Sukhatme. "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem." In Distributed Autonomous Robotic Systems 5, pp. 299-308. Springer Japan, 2002.

[13] P. Song, and V. Kumar. "A potential field based approach to multi-robot manipulation." In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, vol. 2, pp. 1217-1222. IEEE, 2002.

[14] M. M. Zavlanos, and G. J. Pappas. "Potential fields for maintaining connectivity of mobile networks." IEEE Transactions on Robotics, Volume 23, Issue 4, August 2007, pages 812-816.

[15] K. Y. Lee, and J. B Park. "Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages." In Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES, pp. 188-192. IEEE, 2006.

[16] M. S. Couceiro, S. Micael, Rocha R. P., 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, pp. 327-332. IEEE, 2011.

[17] American Electric Lighting, “875 17M 56 FG”, Photometric Data File, 2008.

[18] American Electric Lighting, “AVL 1XM AM DG”, Photometric Data File, 2008.

[19] A. Q. Hoang, and M. T. Pham. "Light Source Detection using Multirobot Systems with Particle Swarm Optimization Approach." In Proceedings of The 3rd Vietnam Conference on Control and Automation, pp. 448-455. Thai Nguyen, 2015.