

Q-Learning Adjusted Bio-Inspired Multi-Robot Coordination

Yan Meng

*Department of Electrical and Computer Engineering, Stevens Institute of Technology
USA*

1. Introduction

In a multi-robot system, each robot needs work together with the network of other robots, considering options for matching its capabilities with demand, negotiating on such constraints as quality, price and time, and then making decisions for committing resources to match demands. Multi-robot systems demand group coherence (robots need to have the incentive to work together faithfully) and group competence (robots need to know how to work together well). With recent advances in all aspects of the technology associated with computing, energy, sensing, and networking infrastructure, more progress has been made in the developing of multi-robot systems for a potentially dynamic, challenging, and hazardous environment. Some examples of such applications include search and rescue, mine detection, hazardous material collection (or cleanup), reconnaissance, smart home/office, surveillance, construction, planetary exploration, and transportation. Since human assistance in these challenging environments is limited due to distance or the need for quick response to changing circumstances, more advanced techniques, such as self-adaptive and self-evolving, would be desirable for these multi-robot systems, which are still unsolved research areas.

Some common challenges exist for this kind of systems. It is often not hard to implement a rudimentary controller that accomplishes the task, but achieving optimal performance can be very challenging. Coordination is hard when robots are really self-interested. If each individual robot is very complex with plenty of intelligent functionalities, through the interacting with other robots and environments, the overall systems will become computational intractable with the large scale agents.

To achieve the global intelligence of a cooperative multi-agent system, distributed coordination methods are more attractive compared to the centralized methods due to its robustness, flexibility, and adaptivity. However, designing a self-adaptive multi-robot system is not a trivial task. Nolfi and Floreano [Nolfi and Floreano, 2000] claim that, since the individual behavior is the emerging result of the interaction between agents and environment, it is difficult to predict which behavior results from a given set of rules, and which are the rules behind an observed behavior. Similar difficulties are present in the decomposition of the organized behaviors of the whole system into interactions among individual behaviors of the system components. The understanding of the mechanisms that led to the emergence of self-organization must take into account the dynamic interactions

among individual components of the system and between these components and the environment. Thus, it is difficult to predict, given a set of individual behaviors, which behavior at the system level will emerge, and it is also difficult to decompose the emergence of a desired global behavior into simple interaction among individuals [Dorigo, 2004]. To develop intelligent robots that can adapt their behaviors based on interaction with the environment and other robots, become more proficient in their tasks over time, and adapt to new situations as they occur, more researchers turned their attentions to bio-inspired systems, such as social insects. Swarm intelligence is an innovative computational and behavioral metaphor by taking its inspiration from the behavior of social insects swarming, flocking, herding, and shoaling phenomena in vertebrates, where social insect colonies are able to build sophisticated structures and regulate the activities of millions of individuals by endowing each individual with simple rules based on local perception.

In this chapter, we propose a novel bio-inspired coordination paradigm, i.e. QVP-PSO, to achieve an optimal group behavior for multi-robot systems, which is the combination of a reinforcement learning method and a bio-inspired Visual Pheromone based Particle Swarm Intelligence (VP-PSO). Basically, two coordination processes among the robots are established in the proposed QVP-PSO architecture. One is a virtual pheromone based algorithm to guide the robots' movements for targets, where each robot has its own virtual pheromone matrix, which can be created, enhanced, evaporated, and propagated to its neighboring robots. The other one is Particle Swarm Optimization (PSO)'s cognitive capabilities through local interaction, which aims to achieve the balance for each robot between the exploration and exploitation through the interactions among the robots using the PSO-based algorithm. To adapt to the changing environment for optimal group behaviors, a Q-learning method is applied to dynamically adjust the associated parameters of the PSO method so that the balance between the explorative, cognitive, and social factors can be optimized under different scenarios.

The paper is organized as follows: Section 2 describes the problem statement. The virtual pheromone based PSO (VP-PSO) scheme is explained in Section 3. Then the Q-learning adjusted VP-PSO method is discussed in Section 4. Section 5 presents the simulation results using the embodied robot simulator: Player/Stage. To conclude the paper, Section 6 outlines the research conclusions and the future work.

2. Related work

Extensive multi-robot coordination techniques have been developed for various applications, such as foraging, box-pushing, aggregation and segregation, formation forming, cooperative mapping, soccer tournaments, site preparation, sorting, and collective construction. [Balch and Arkin, 1999] [Stewart, 2006] [Dias et al., 2004] [Burgard et al., 2005] [Chaimonwicz, 2004] [Fernandez, 2005][Martinoli, 1999] [Weigel, 2006] [Parker and Zhang, 2006] [Holland and Melhuish, 1999][Correll and Martinoli, 2006]. All of these systems consist of multiple robots or embodied simulated robots acting autonomously based on their own individual decisions.

Some computational intelligence algorithms have been proposed, such as neural networks [Rumelhart and McLelland, 1986], genetic algorithms [Holland, 1975], evolution strategies [Rechenberg, 1973], immune networks [Bersini and Varela, 1991], Ant Colony Optimization

(ACO) [Dorigo et al, 1996] and Particle Swarm Optimization (PSO) [Kennedy and Eberhart, 1995], seeking to replicate the related natural behaviors. All of the above nature-inspired computational intelligence has proved to be effective and efficient approaches towards design and control of complex problems under dynamic environments.

Although artificial evolution has been often used for synthesizing behaviors for autonomous robots [Nolfi and Floreano, 2000], its use as a methodology to evolve behaviors for groups of robots is still limited. Reynolds [Reynolds, 1987] built a computer simulation to model the motion of a flock of birds, called *boids*. He believes the motion of the *boids*, as a whole, is the result of the actions of each individual member that follow some simple rules. Ward et al. [Ward, 2001] evolved *e-boids*, groups of artificial fish capable of displaying schooling behavior. Spector et al. [Spector et al., 2003] used genetic programming to evolve group behaviors for flying robots in a simulated environment. The above mentioned works suggest that artificial evolution can be successfully applied to synthesize effective collective behaviors. Dorigo et al. [Dorigo et al., 1996] developed a robotic system consisting of a swarm of *s-bots*, mobile robots with the ability to connect to and to disconnect from each other depends on different environments and applications, which is based on behaviors of ant systems. Payton et al. [Payton et al., 2001] proposed pheromone robotics, which was modeled after the chemical insects, such as ants, use to communicate. Instead of spreading a chemical landmark in the environment, they used a virtual pheromone to spread information and create gradients in the information space. By using the virtual pheromone, the robots can send and receive directional communications to each other. Pugh and Martinoli [Pugh and Martinoli, 2006] proposed a group learning algorithm using Particle Swarm Optimization for a multi-robot system, where aggregation behaviors were conducted to evaluate the proposed methods. Werfel and Nagpal [Werfel and Nagpal, 2006] proposed an extended pheromone by increasing the capabilities of environmental elements in swarm robots to automatically assemble solid structures of square building blocks in two dimensions according to a high-level user-specific design. In our previous work [Meng et al. 2007][Meng and Gan, 2007], swarm intelligence based robot coordination methods were proposed.

3. Virtual pheromone based PSO method

The objective of this study is to design an efficient and robust distributed coordination algorithm for a multi-robot system with limited on-board power, sensing and communication on each robot, aiming at optimizing the group behavior especially for a searching task under a dynamic environment. The targets can be defined as any kind of predefined object. It is assumed that the searching area is bounded and robots can detect the targets using special on-board sensors, such as camera systems. The robot can only detect the targets within its local sensing range. Once a robot detects a target, it processes the target assuming the processing time is proportional to the size of target. Assume that the robots are simple, and homogeneous. Each robot can only communicate with its neighbors. Two robots are defined as neighbors if the distance between them is less than a pre-specified communication range. The goal is to detect and process all of the targets within the searching area as soon as possible.

3.1 Virtual Pheromone

Pheromone is a class of mechanisms that mediate animal-animal interactions through artifacts or via indirect communication, providing a kind of environmental synergy, information gathered from work in progress, distributed incremental learning and memory among the society. To emulate pheromone-based communication in a multi-robot system, special pheromone materials and associated detectors need to be designed, and most of the time such chemical/physical pheromone is unreliable and easily be modified under some hazardous environments, such as urban search and rescue. A modification of this autocatalysis is necessary. Similar to [Payton, 2001], a unique virtual robot-to-robot interaction mechanism, i.e. *virtual pheromone*, was proposed as the message passing coordination scheme for the swarm robots.

Each target in the environment is associated with one unique pheromone, which can be enhanced or evaporated over time to adapt to a dynamic environment. Initially, each robot creates its own virtual pheromone matrix, which installs all pheromone information associated with different targets. Whenever a robot detects a target, it would update its own pheromone matrix and broadcast this target information to its neighbors through a visual pheromone package.

3.2 Fitness function

To emulate the pheromone creation, enhancement and elimination procedure in natural world, the pheromone density $\tau_{ij}^k(t)$ can be updated by the following equation:

$$\tau_i^k(t+1) = \rho(\tau_i^k(t) + \alpha) - (1 - \rho)m\tau_i^k(t) \quad (1)$$

where $\tau_i^k(t)$ represents the pheromone density of target i at time t for agent k . $0 < \rho < 1$ is the enhancement factor of the pheromone density. α is the pheromone interaction intensity received from the neighboring robots for target i . Basically, α is used for pheromone enhancement, and m represents the elimination factor. In the ants system, the pheromone will be eliminated over time if it is not being enhanced by the ants, and the elimination procedure usually is slower than the enhancement. When the pheromone trail is totally eliminated, it means that no target is needed to be processed through this pheromone. To slow down the elimination relative to enhancement, m is set as less than 1.

To define the probability that robot k moves toward target i with pheromone density $\tau_i^k(t)$, the target utility function is defined as following:

$$\mu_i^k(t) = \tau_i^k(t)e^{-\gamma_i} \quad (2)$$

where γ_i represents local target redundancy, which is defined as the number of the local neighbors who have sent the pheromone referring to the same target i to robot k .

Generally speaking, the higher the target utility is, the more attractive the corresponding target is to the robot. Therefore, the benefit of moving to this target would be higher in terms of the global optimization. If the local target redundancy is high, it means that there will be more potential robots (globally) moving to this target, which may lead to the less available targets left in the future. Therefore, the benefit of moving to this target would be less in terms of the global optimization. With the local redundancy, we are trying to prevent

the scenarios that all of the robots within a local neighbor move to the same target instead of exploring new targets elsewhere.

Initially, the robots are randomly distributed in the searching environment, where multiple targets with different sizes and some static obstacles are randomly dispersed within the environment. At each iteration, if each robot adjusts its behavior based only on the target utility, it may lead the robot to be very greedy in terms of the robots' behaviors, since the robots would rather move to the target with higher utility than explore new areas. This greedy behavior of the robots may easily lead to local optima.

To prevent the local optima scenarios with utility-greedy method using (2), the target visibility has to be considered as well. Let $\eta_i^k(t)$ denotes the target visibility for agent k in terms of target i , which is defined as:

$$\eta_i^k(t) = \max\left(\frac{\delta^k}{d_i^k(t)}, 1\right) \quad (3)$$

where δ^k represents the local detection range of robot k , and the $d_i^k(t)$ represents the distance between the robot k and target i . When the target visibility is higher, it means the distance between the target and the robot is smaller, it would be more beneficial to move to this target due to its lower cost compared to moving to the more distinct target under the same environmental condition.

3.3 Coordination of Robot Behaviors

One of the objectives of the pheromone update rules is to prevent stagnation, which occurs when most of the robots follow the same path, or converge to the same target. In general, global updates will facilitate exploitation, while local updates will favor exploration by letting each robot update pheromone after each transition decision. During each local update, the pheromone will diminish due to evaporation. Over time, frequently visited links will become less attractive, thus favoring exploration of less frequently used links. The local update by each agent therefore will avoid a very strong link from dominating as a component of the final solution. The utility-greedy using (2) has the tendency which may lead to the stagnation.

Now the question is how to integrate the target utility and target visibility into an efficient fitness function to guide the movement behaviors of each robot so that the stagnation can be avoided. Stagnation occurs when most of the robots follow the same path, or converge to the same target. To tackle this issue, we turned our attention to the Particle Swarm Optimization (PSO) method. The PSO algorithm is a population-based optimization method, where a set of potential solutions evolves to approach a convenient solution (or set of solutions) for a problem. The social metaphor that led to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by every possible individual. Individuals may modify this "opinion state" based on three factors: (1) The knowledge of the environment (explorative factor); (2) The individual's previous history of states (cognitive factor); (3) The previous history of states of the individual's neighborhood (social factor).

Basically, the PSO algorithm can be represented as in (4), which is derived from the classical PSO algorithm [Kennedy and Eberhart, 1995] with minor redefinitions of formula variables:

$$v = \textit{explorative} + \textit{cognitive} + \textit{social} \quad (4)$$

where v is the velocity of a robot. To determine which behavior is adopted by robot k , the velocity has to be decided first. If the received pheromone density is high, the robot would increase the weight of social factor, and decrease the weight of the cognitive and exploration factors. On the other hand, if the local visibility is of significant to the robot, then the velocity of the robot would prefer the cognitive factor to the social factor. If both social and cognitive factors are low, then the exploration factor would be increased. Furthermore, at any given time, the velocity of the robot would leave some spaces for the exploration of new areas no matter what. Therefore, the basic idea is to propel towards a probabilistic median, where explorative factor, cognitive factor (local robot respective views), and social factor (global swarm wide views) are considered simultaneously and try to merge these three factors into consistent behaviors for each robot. The exploration factor can be easily emulated by random movement.

The challenging part is how to define the local best (cognitive factor) and the global best (social factor). One straight forward method is to select the highest target visibility from a list of available targets as the local best. If only one target is on the list, then this target would be the local best. The easy way to select global best is to select the highest target utility from a list of available targets.

Instead of defining a fitness function, for a robot system, the robot velocity vector including both magnitude and direction would be a better representation to control the movement behavior. Based on the above discussion and the PSO algorithm, each robot would control its movement behaviors by following this equation:

$$v^k(t+1) = \psi_e \text{rand}_e(\cdot) v^k(t) + \psi_c \text{rand}_c(\cdot) (p_c - x^k(t)) + \psi_s \text{rand}_s(\cdot) (p_s - x^k(t)) \quad (5)$$

where, ψ_e , ψ_c , and ψ_s represent the propensity constraint factors for explosive, cognitive, and social behaviors, respectively, $0 \leq \text{rand}_\Theta() < 1$ where $\Theta = e, c, \text{ or } s$, and $x^k(t)$ represents the position of robot k at time t . $p_s = \max(\mu_i^k(t))$ represents the global best from the neighbors, and $p_c = \max(\eta_i^k(t))$ represents the local cognitive best. The position of each robot k at time $t+1$ can be updated by

$$x^k(t+1) = x^k(t) + v^k(t+1). \quad (6)$$

4. Q-Learning adjusted method

Since the environment is dynamic, it is difficult to guarantee that the initial parameters of the VP-PSO method will be the best-fit for the current scenario all the time. Therefore, it is necessary to dynamically adjust these parameters to achieve optimal global performance and expedite the convergence.

After some preliminary experiments using the VP-PSO method, it was observed that with different weight parameters of the PSO, the searching performance varied noticeably, i.e., the convergence rate and the target/robot distribution ratio may be different. For example, it is assumed that there are 10 targets which are distributed in groups in three different locations. There are 20 robots. We conduct 2 cases with different parameters of the PSO method. In case 1, the target/robot distribution map is 3(target)/10(robot), 5/6, and 2/4. In case 2, the distribution map becomes 3/7, 5/12, and 2/1. In case 1, after finishing the 3

targets, the first group of robots has to start searching again for other unfinished targets, such as second locations. Extensive travel cost would be consumed for the searching. In an ideal situation, each target should attract 2 robots by average. The processing time could be reduced significantly since extensive unnecessary travel cost has been saved. However, since the system is distributed, there is no global station to check the status of other targets or robots. Each robot has to make decisions based on its local sensor feedbacks.

It would be desirable that robots have the self-learning capability to dynamically adjust the parameters of the coordination algorithms according to their current status and sensor feedbacks. Since the environment of a multi-robot system is dynamically changing, the measurement feedbacks from the environment would be critical to help to adjust the coordination methods for the changing situations. To achieve a higher learning performance, it is assumed that the predefined expected robot/target ratio is provided. Q-learning method is applied to adjust the parameters of the PSO for better coordination behaviors. Q-learning is a learning technique for acquiring optimal actions based on the evaluation values $Q(s,a)$ (Q-value) for state-action sets. To simplify the method, the gradient-based action for parameter adjustment will be applied. For example, parameter X will be increased by 0.5 unit of gradient and Y will be decreased by 1.3 unit of gradient. The Q-learning method is summarized as followings:

1. Initialize all states $s \in S$ and action $a_i \in A_i$, and $Q_i(s, a_1 \cdots a_n) = 0$;

2. Repeat for every 10 step's state $s \in S$:

1) Get target info $w_i(t)$ from the pheromone matrix;

2) Choose an action a_i according to $a_i = \arg \max_{a_i \in A_i} Q_i(s, a_1^t \cdots a_n^t)$

3) Observe rewards r_i^t , and the next state S^{t+1}

$$Q_i(s^{t+1}, a_1^{t+1} \cdots a_n^{t+1}) = (1 - \beta)Q_i(s^t, a_1^t, \dots, a_n^t) + \beta(r_i^t + \lambda \tilde{Q}_i(s^{t+1}, a_1^{t+1}, \dots, a_n^{t+1}))$$

$$\text{where } \tilde{Q}_i = p_i \max \left(\frac{w_i^t - \varepsilon \gamma_i^t}{d_i^t}, 0 \right) + (1 - p_i) \frac{(w^T - w_i^t)}{\text{rand}(t)(d_i^t - \delta)}$$

where β is the learning factor. w_i^t is the weight of detected target i . γ_i^t represents the estimated value of robots redundancy at target i , ε is the expected target/robot ratio, and d_i^t is the distance between the robot's current position and the target i . Therefore, $\frac{w_i^t - \varepsilon \gamma_i^t}{d_i^t}$

represents the benefit a robot can gain by moving towards target i . When the number of robots around a target extends the expected target/robot ratio, the benefit is set to 0. w^T is the total target weights within the search area, and δ is the sensor detection range. It is assumed that the targets are distributed within the search area in uniform probability

density, thus the benefit of random search can be represented by $\frac{(w^T - w_i^t)}{\text{rand}(t)(d_i^t - \xi)}$. p_i is the

probability of going to target i , which is defined as $p_i = \frac{\psi_s}{\psi_e + \psi_c + \psi_s}$, where $\psi_e, \psi_c,$

and ψ_s are the weight parameters of Equation (5). The reward $r_i^t = 1$ if the robot number around the target i is approximately equal to ε , otherwise, $r_i^t = -1$.

5. Experimental results

5.1 Experimental setup

Player/Stage is selected as our embodied robot simulator to implement the QVP-PSO algorithm using swarming robots. As shown in Fig. 1, the environment is an open space with the size of 41.8m x 45.1m, where several targets are distributed randomly. 20 homogeneous Pioneer 3DX robots are used as our robot model, which is equipped with a camera system to detect and track targets, a laser range finder to measure the distance between the target and itself, a sonar sensor to avoid obstacles (i.e. both static obstacles and mobile obstacles, such as other robots), and a wireless communication card to communicate with other robots. The arc shape in front of each robot represents the field of view of the vision system on each robot. The communication range is set up as the same range of the vision but using a circle instead of an arc. Whenever the robots are within other robot's communication range, they would exchange the information between them.

5.2 Robot localization and path planning

Self-localization is critical for multi-robot to coordinate with each other. Since the robots are working in an unknown environment, an a priori map is not available. Building a map using distributed multi robots is a challenging task requiring more computational complexity, which is not the focus of this paper. Since it is assumed that each robot is simple and small with limited on-board battery, a simplified localization is required. It is assumed that the initial positions of all robots relative to a global coordinate frame of searching area are given. Since each robot has encoders attached with wheel motors, odometry-based localization is employed here.

Since the targets are designed with different colors, a color blob detection using on-board camera system is carried out for target detection. Once a robot detects one or more targets, it uses on-board laser range finder to estimate the distance of the target relative to its own position. Then the target information can be propagated to its neighbors with the target position. When a robot picks a target, it would track the target color using the vision system while moving toward the target.

There are two important factors contributing to a dynamic environment. One is that robots always have to avoid obstacles, including static obstacles (i.e. walls) and dynamic obstacles (i.e. other mobile robots). It would be too computationally expensive for a robot since the global path has to be recalculated every time the robot avoids an obstacle. On the other hand, once a robot turns around to avoid one obstacle, it may become closer to other targets or it may receive new pheromone from its new neighbors, which naturally leads to new path planning for this robot. Therefore, it would be inefficient to use any complex global path planning. Here, a simplified path planning method is employed, where a robot sets up a destination location based on the detected or received target information, then the robot moves directly toward this destination. If an obstacle is in its way, the robot turns 45 degrees left if the obstacle is on the right side or turns right if the obstacle is on the left to

avoid the obstacle, then continue moving towards the destination. The destination may be changed due to new detection or new pheromone.

5.3 Experiments under an open-space environment

An open space experiment is conducted as shown in Fig. 1. Initially, the robots are randomly searching for targets at $t = 1$. Once a robot detects a target, it would propagate the pheromone of this target to its neighbors, as shown in Fig. 1(b), where a small rectangle beside a robot indicates that the on-board vision system has detected the targets. After receiving a pheromone message, robots make their own movement decisions based on the QVP-PSO algorithm, as shown in Fig. 1(c), Fig. 1(d), and Fig. 1(e). The simulation stops when all of the targets have been found and processed, as shown in Fig. 1(f).

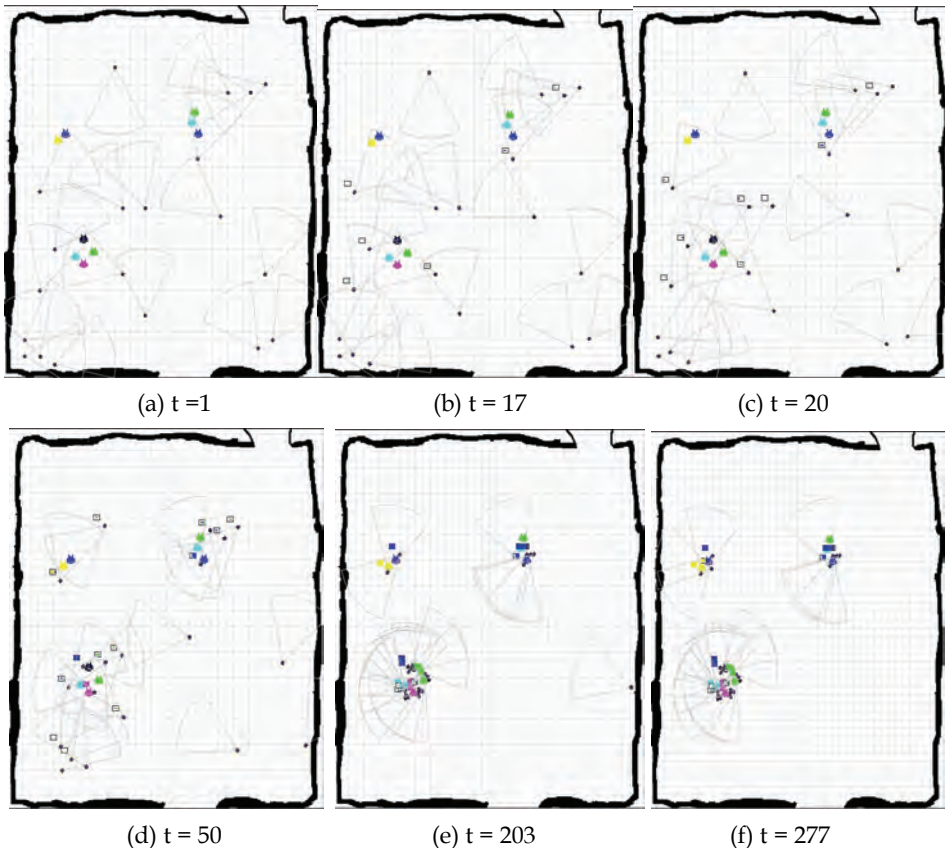


Figure 1. 20 robots search for randomly distributed targets in an open space on a player/stage simulator at $t = 1, 17, 20, 50, 203,$ and 277 time steps

To evaluate the robustness of the QVP-PSO algorithm under a dynamic environment, another set of experiments are conducted in an open space, as shown in Fig.2. Since it is not allowed to dynamically change the target configuration in Player/Stage, the target relocations are conducted manually. As shown in Fig. 4, initially, 20 robots search for

targets. After some robots have reached to a target, the target is manually relocated to somewhere else. All of the robots around this target would disperse to explore new areas for new targets until all of the robots converge to the targets. It can be seen that the QVP-PSO algorithm is very robust to adapt to the unexpected events, such as target relocation.

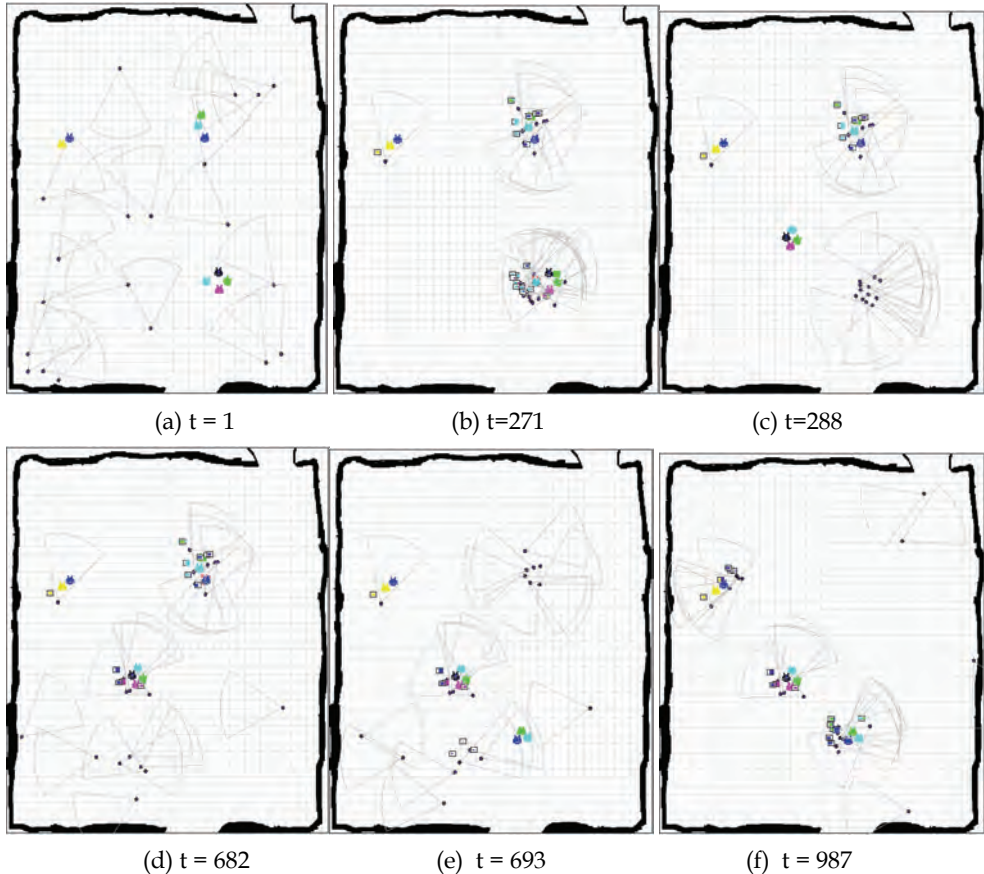


Figure 2. Snapshots of simulation in an open space on Player/Stage with dynamic target relocations. (a) initial state; (b) all robot converge to targets; (c) relocate targets; (d) converge to new targets; (e) relocate targets; (f) converge to new targets

5.4 Experiments under an indoor office environment

Second set of experiment is conducted in an indoor office environment, as shown in Fig. 3, where the size of environment is 41.8m x 45.1m. Several targets with different colors are randomly distributed in the environment, and 20 robots search for them. The major difference of this experiment compared to the open space lies in the fact that even if the robot received the target information from its neighbors, it may not be able to move toward the target if there is a wall exits between the robot and target. The robot may have to spend some time to avoid the wall, and during this procedure, if new target information comes in, the robot may change its mind to move forward to new target based on the QVP-PSO algorithm.



Figure 3. 20 robots search for randomly distributed targets in an indoor office environment on a player/stage simulator at $t = 1, 13, 15, 58, 423,$ and 495 time steps

5.5 Experimental Results

To evaluate the performance of the QVS-PSO algorithm, two other methods are carried out for comparison. One is random movement (Random), where all robots search for targets in a random manner without communicating with others. Second one is the VS-PSO algorithm without Q-learning adjustment. To obtain the statistic performance, we implemented the following experiments. 10 targets are distributed in the environment with fixed positions for the simulations. Then, we start running the simulations with the swarm size of 20 using three methods, each method runs 35 times to obtain result of ending time, the total travel time of all robots, and the mean squared error of robot/target distribution ratio. The results are shown in Fig. 4, Fig.5, and Fig.6, respectively.

The search ending time represents the searching performance. The less ending time the system takes, the faster the robots detect and process all the targets. To take the overall power consumption into the considerations, the total travel time of all the robots is a good measurement since robot movements usually consume higher percentage of power compared to the power consumption for the communication and onboard computation processing of the robots. The robot/target distribution ratio errors could tell us how good the algorithm can achieve to dynamically allocate the robots into different targets dynamically in a more reasonable manner.

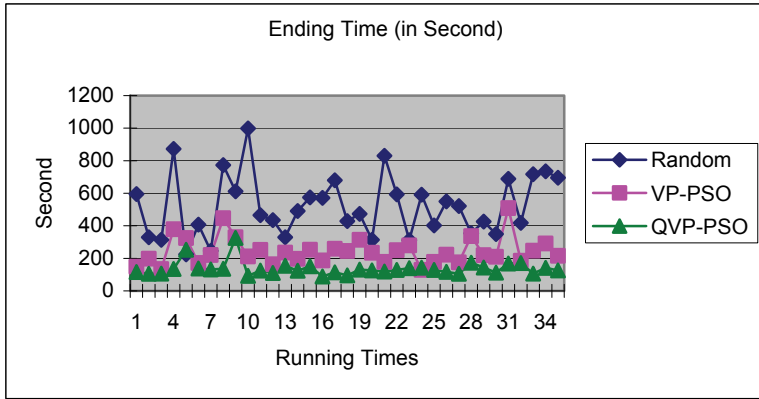


Figure 4. The search ending time vs. run times

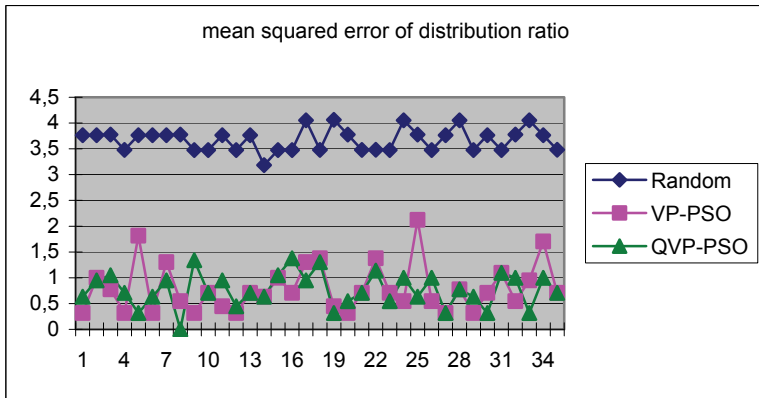


Figure 5. The mean squared error of the robot/target distribution ratio vs. run times

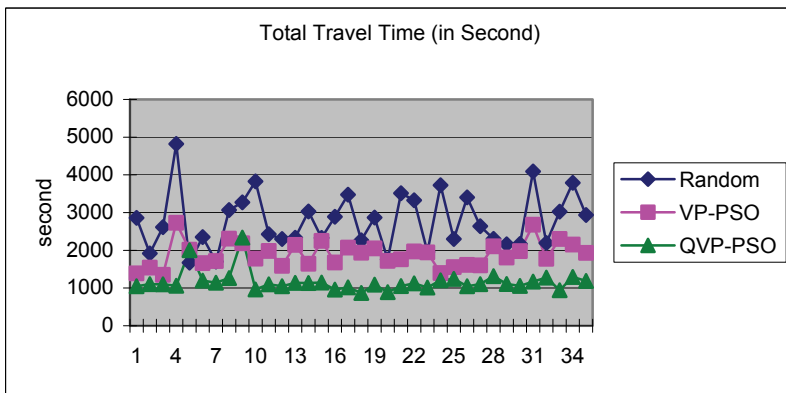


Figure 6. Total travel time of all robots vs. run times

It is obvious that the random approach takes much longer time than the other two approaches and obtains the bigger error in the robot/target distribution ratio. Although sometimes the QVP-PSO algorithm and VP-PSO algorithm have the similar performance under specific target configuration, overall the QVP-PSO algorithm outperforms the VP-PSO algorithm in the search ending time, the total travel time, and the robot/target distribution errors under different target configurations.

6. Conclusion

The proposed QVP-PSO algorithm has the following characteristics: (1) Robots act independently, asynchronously, and in parallel, without maintaining a global model; (2) Robots use a simple control algorithm regardless of the changes under a dynamic environment; and (3) Robots can only communicate with their neighbors to share information; (4) the randomness of the robot movement has been reduced to achieve more reasonable robot/target distribution. Compared to the VP-PSO method, extensive simulation results demonstrate the higher performance of the QVP-PSO algorithm in three different measurements, i.e., search ending time, total travel time of robots (which is directly related to the overall power consumption of the system), and the robot/target distribution ratio errors.

However, due to the dynamic characteristics of this distributed multi-robot system, it is difficult to estimate the target utility value with high accuracy because each individual robot makes its own decision independently based on its local view and some neighboring view. The robot sensor may also bring the noise for the target detection. Probability approaches tend to be more robust in the face of sensor limitation and model limitations. To improve the target detection rate and system robustness, the probability-based approaches will be investigated in the future.

7. References

- Balch, T & Arkin, R. C. (1999). Behavior-based Formation Control for Multi-robot Teams, *IEEE Trans. on Robotics and Automation*.
- Bersini, H. and Varela, F.J. (1991), The immune recruitment mechanism: a selective evolutionary strategy. In *Proc. Fourth Int. Conf. Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 1991, pp.520-526.
- Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated Multi-Robot Exploration, *IEEE Trans. on Robotics*, Vol. 21, No. 3.
- Chaimowicz, L.; Kumar, V., and Campos, M. F. (2004). A Paradigm for Dynamic Coordination of Multiple Robots, *Autonomous Robots*, Volume 17, Issue 1, July 2004: pp. 7-21.
- Correll, N. and Martinoli, A. (2006). System Identification of Self-Organizing Robotic Swarms, in *the 8th Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*, Distributed Autonomous Robotic Systems, pp. 31-40, Springer Verlag, 2006.
- Dias, M.; Zinck, M., Zlot, R. and Stentz, A. (2004). Robust Multirobot Coordinate in Dynamic Environments, in *Proceedings of IEEE International Conference on Robotics and Automation*, 2004, pp.3435 - 3442.
- Dorigo, M.; Maniezzo, V., and Colormi, A. (1996). Ant System: Optimization by a Colony of Cooperating Robots, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 26, No. 1, February 1996.
- Dorigo, M.; Trianni, V., Sahin, E. (2004). Evolving Self-Organizing Behaviors for a Swarm-bot. *Autonomous Robots*, Vol. 17, No.2-3, 2004. pp. 223-245.

- Fernandez, F., Borrajo, D., and Parker, L. E. (2005). A Reinforcement Learning Algorithm in Cooperative Multi-Robot Domains, *Journal of Intelligent and Robotic Systems*, vol. 43, nos. 2-4, 2005: 161-174.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.
- Holland, O.E. and Melhuish, C. (1999). Pheromone, Self-Organization, and Sorting in Collective Robotics, *Artificial Life*, Vol. 5, pp. 173-202, 1999.
- Kennedy J. and Eberhart, R. (1995). Particle Swarm Optimization, *IEEE Conference on Neural Networks*, Proceedings 1995.
- Martinoli, A.; Ijspeert, A. J. and Mondada, F. (1999). Understanding Collective Aggregation Mechanisms: From Probabilistic Modeling to Experiments with Real Robots, *Robotics and Autonomous Systems*, Vol. 29, pp. 51-63.
- Meng, Y.; Kazeem, O., and Muller, J. (2007). A Hybrid ACO/PSO Control Algorithm for Distributed Swarm Robots. *2007 IEEE Swarm Intelligence Symposium*, Hawaii, USA.
- Meng, Y. and Gan, J. (2007). LIVS: Local Interaction via Virtual Pheromone Coordination in Distributed Search and Collective Cleanup. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, Oct. 29-Nov.2, San Diego, CA, USA.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA.
- Parker, C.A.C. and Zhang, H. (2006). Collective Robotic Site Preparation. *Adaptive Behavior*. Vol.14, No. 1, 2006, pp. 5-19.
- Payton, D.; Daily, M., Estowski, R., Howard, M., and Lee, C. (2001). Pheromone Robotics. *Autonomous Robots*, Vol. 11, No. 3.
- Pugh, J. and Martionli, A. (2006). Multi-robot learning with particle swarm optimization, *Proceedings of the Fifth International Joint Conference on Autonomous Robots and Multirobot Systems*, AAMAS, 2006, Hakodate, Japan, pp. 441-448.
- Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4), July 1987, pp. 25-34.
- Rechenberg, I. (1973). *Evolutionsstrategie*. Stuttgart: Fromman-Holzboog.
- Rumelhart, D. E. and McLelland, J. H. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press.
- Spector, L.; Klein, J., Perry, C., and Feinstein, M. D. (2003). Emergence of collective behavior in evolving populations of flying robots. In E. Cantu-Paz et. al. editor. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, Berlin, Germany, 2003, pp.61-73.
- Stewart, R. L. and R. A. Russell. (2006). A Distributed Feedback Mechanism to Regulate Wall Construction by a Robotic Swarm. *Adaptive Behavior*. 14(1):21-51.
- Ward, C. R.; Gobet, F., and Kendall, G. (2001). Evolving collective behavior in an artificial ecology. *Artificial Life*, Vol. 7, No. 2, 2001, pp.191-209.
- Weigel, T.; Gutmann, J. S., Dietl, M., Kleiner, A., and Nebel, B. (2002). CS Freiburg: Coordinating Robots for Successful Soccer Playing. Special Issue on Advances in Multi-Robot Systems, T. Arai, E. Pagello, and L. E. Parker, Editors, *IEEE Trans. on Robotics and Automation*, Vol. 18, No.5, pp. 685-699.
- Werfel, J. and Nagpal, R. (2006). Extended Pheromone in Collective Construction. *IEEE Intelligence Systems*, Vol. 21, No.2, pp. 20-28.