Multiple Robot Coordination for Surveillance Task Using Market Approach

Wen-Chieh Fang, Li-Hui Chen, Yu-Ming Liao, Bing-Shen Ho, and Jane Yung-jen Hsu
Intelligent Agents Group, iSpace Laboratories
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
wcf123@ntu.edu.tw, b91065@csie.ntu.edu.tw, dycan0716@gmail.com, b91902063@ntu.edu.tw, and yjhsu@csie.ntu.edu.tw

Abstract

The keys to utilizing the potential of multi-robot systems are cooperation and coordination. Robots nowadays are equipped with better computing, high-bandwidth communication, and many kinds of sensors and actuators. Designing an effective robot team to achieve a global task is a big challenge. In addition, robots will face more uncertainties in an unknown or dynamic environment. The main investigation of this paper focuses on multiple robot coordination based on a market approach. In simulation experiment domain, a primitive simulation framework is built based on Webots software. The scenario can be divided into three technical domains: deadlock domain, specific uncertainty domain, and fewest robot strategy vs. best efficiency strategy domain. This paper demonstrates that, to achieve surveillance task, a group of robots can collaborate to detect emergencies in an indoor environment.

Keywords: Multi-robot system, market-based approach.

1. Introduction

This paper copes with a task allocation problem in a group of autonomous robots with different capabilities. In biology, swarm insects such as ants or bees can work together to achieve a goal. Higher-level animals like wolves often hunt in a group. Can a robot team have such fantastic behaviors or intelligence? In software agent systems, *multi-agent system* and *distributed artificial intelligence* develop a lot of promising results. Although there is much research in negotiation among software agents, directly using these approaches to solve the problems in robotics is questionable [7]. A significant difference between software agents and robotic agents is that the latter often deal with more limited capabilities such as more restricted communication. Moreover, various failures occur more frequently in robotic systems. For instance, in order to adapt to faulty perception, robotic systems have to be able to accommodate larger error bounds. Thus, many real-world factors should be taken into consideration in robotic systems.

One of the most important issues that affect the architecture and operation of the robot team is heterogeneity. Heterogeneous teams are those in which there are at least two agents with different hardware or software capabilities. Currently, the main disadvantage of homogeneous robot teams is that they can only tackle very simple tasks. Therefore, heterogeneous teams are used to perform more complex tasks. In addition, designing robots with a small set of skills is often easier than designing fully capable robots.

This paper proposes a market-based approach to the coordination of a heterogeneous robot team. Many potential robotic capabilities exist and this paper explicitly assumes that robots will have different capabilities. Each complex task needs different capabilities so that the task can be dealt with effectively and correctly. This paper chooses the capabilities by auction, and illustrates our auction mechanism through a simulated system, which is built in Webots 3D software platform [20].

The rest of this paper is organized as follows. In next section a brief survey of related research on this problem is presented. In Section III, the multi-robot surveillance problem is defined and the concept of capability representation is proposed. The whole implementation is described in Section IV. The simulation scenarios and related results and discussion are shown in Section V. Section VI is the conclusion.

2. Related Work

The problem of multi-robot coordination in our research is related to many research domains, here the focus is on cooperative surveillance, task allocation and a relevant solution, i.e. market-based approach.

2.1 Cooperative Surveillance

A typical surveillance mission is an application of robot coverage problems [2]. In [6], the robot coverage problem is divided into three categories: blanket coverage, barrier coverage, and sweep coverage. Our surveillance task is very similar to sweep coverage because, during a particular time and in a specific area, the *search* robots try to maximize the number of detections and minimize the number of misses.

The surveillance/search task can also be considered as a dynamic coverage problem because the environment is so large that every point in the environment can not be under the robots' sensor shadows at each moment of time. Therefore, robots must thus randomly move in order to observe as many points in the environment as possible.

In addition, our surveillance task is different from canonical clean-floor task. The canonical clean-floor task places emphasis on keeping track of the areas already cleaned. However, this is not important for the surveillance task because emergencies may happen unexpectedly anywhere.

In this paper, each robot for a surveillance task is also like a mobile sensor. In [19], a novel system called *Multi-Robot Sensor Network* (MRSN) is proposed. It has two key features that are not observed in MANETs (mobile ad hoc networks): disconnection/delay tolerance and intentional mobility. The disconnection tolerance is allowable in terms of the existence of intentional mobility. That is, disconnection of communication links among sensor nodes can be compensated for by the intentional reach of the mobile sensor node toward the communication area of the destination node.

In [17], optimal sensor placement and target location problem are described. Extension of mobile sensors is also introduced. Two ways to deploy sensors are deterministic approach and self-organizing approach. For a highly dynamic environment, the self-organizing approach is preferred because the sensing location will change frequently. Dynamic deployment can respond to the change of the sensed environment in a real-time manner. Traditional sensor networks always pay attention to the positions of the sink or cluster-head from the viewpoint of energy efficiency and performance.

In [15], each member, of the *Active Sensor Networks* (ASNs), has the global synchronized world view. In contrast, the nodes of a traditional sensor network may not have the global knowledge and may just perform the sensing task. The entities of ASN have more comprehensive capabilities than those of traditional sensor networks.

2.2 Task Allocation

In [10], there are two different meanings of the term *task allocation*. The first meaning is defining optimal number of robots necessary to perform the given task. The second meaning is of the assignment of robots to tasks.

The problem in our research corresponds to the first description of task allocation. In other words, it can be defined as the *dynamic resource allocation problem* which is proposed in [8].

2.3 Market-Based Approach

Auction-based or market-based multi-robot coordination approaches received noticeable attention in recent years and were successfully implemented in a variety of multi-robot research domains. Readers can refer to a new review by M. Bernardine Dias et. al. [4] which surveys the state of art in the field.

This paper is most similar to [8] which proposes an auction-based task allocation system called MURDOCH. It is built upon a principled, resource centric, *publish/subscribe* communication model. MURDOCH produces a distributed approximation to a globally optimum resource usage. Finally, they validated MURDOCH in two different domains : a tightly coupled multi-robot manipulation task and a loosely coupled multi-robot experiment in long-term autonomy.

Auctions where bidders submit bids on combinations (called combinatorial auctions) recently received much attention in the multi-robot field [14] [5]. In [5], two types of auctions were compared: simple auction and combinatorial auction. The experiment`s results show that combinatorial auctions have higher scores in simulated games. Inspired from combinatorial auctions, [14] proposed a *combinatorial bids based mechanism*. In our research, because the bidders may bid on bundles of capabilities required for a task, our approach has an implicit relationship with combinatorial auctions. Therefore, combinatorial auctions are a future relevant research area for us.

3. Problem Statement

3.1 Problem Definition

The *multi-robot surveillance problem* is defined as follows.

Given

- A set of robots *R*, each is identified by a geographic location and capabilities, and
- A sequence of tasks *T*, each task *t* is identified by an occurrence time, an expiration time, a geographic location, and resource requirements.

Then the problem is to find an assignment of robots to the tasks, such that

- Each task *t* in *T* is assigned a subset of robots in *R* with sufficient capabilities to meet its resource requirements.

The robots are assumed to be general-purpose mobile platforms. They are honest and cooperative. Moreover, a robot may break down any time, and may not be aware of its own failure.

There are static obstacles in the robot environment. Tasks show up in random locations at random times and there are multiple tasks at each time. A task is considered done as long as the capable robot team arrives at the task, and this task is denoted a successful task. On the other hand, the task is a failure if at least one member of the capable robot team arrives too late (beyond the task expiration time). When the resource requirement of a task is satisfied by more than one robot team, the system needs a strategy to decide the team assignment.

Two performance metrics are used to evaluate the performance of different strategies.

- Task completion time: the time required for a capable robot team to reach the task.
- Task success rate: the ratio of the number of successful tasks to that of total tasks.

Our coordination strategies consider two different *fitness* variables.

- The number of robots in a capable team.
- The distances between each member of the team and the task.

In general, an optimal strategy should minimize task completion time and maximize task success rate. But the choice of strategies is a trade-off. Section V-C gives a further discussion.

3.2 Capability Representation

Because specific capabilities of robots can be used to tackle different tasks, this paper assumes that there is a mapping between the capabilities of robots and the resources that tasks need. If any collection of robots matches the mapping of a task, the robot team is assumed to perform the task independently and effectively. For example, if a task T needs resources: $\{w_1S_1, w_2S_2\}$, where S_1, S_2 are the resource units, and w_1, w_2 are the weights, and if a robot R_1 has capabilities that map to resources such as: $\{w_1^{R1}\ S_1\}$ and another robot $R2$ has capabilities that map to resources such as: $\{w_1^{R2}\ S_1,\ w_2^{R2}\ S_2\}$, where w_1^{R2} and w_2^{R2} are the weights. Then, if robot R_1 and robot R_2 cooperate with each other, we represent their global capabilities as: $\{(w_1^{R1} + w_1^{R2})\ S_1,\ w_2^{R2}\ S_2\}$. If these weights satisfy $(w_1^{R1} + w_1^{R2}) >= w_1$,and $w_2^{R2} >= w_2$, we say the task T can be dealt with by robot R_1 and R_2.

In our simulation, without loss of generality, we assume all weights are one.

4. Implementation

The features of our system are as follows:

1) Our approach presents a new auction based coordination mechanism by extending the help-based cooperation protocol [15].
2) This paper uses multi-level agent architecture to organize the operation architecture from team-level decision-making to low-level motion controlling.
3) Given the coordinates of the task, the robots use numerical potential field method to reach the target.
4) The system can handle failure and uncertainty situations. Section V demonstrates this feature.

4.1 Agent Architecture

In robotic agent architecture, there are three layers: Control layer, Strategy layer, and Team layer. See Fig.1.

1) Control layer: This layer contains perception module and motion control module. This layer is responsible for short-term decision-making, such as taking data from sensors, sending data to the robot strategy layer, and controlling the robot's wheels. The robot has an obstacle-avoiding mechanism while moving toward the target or doing random walk.

For simplicity and robustness, this paper uses Braitenberg architecture [3] to make robots move randomly and avoid obstacles. Each robot has two motors left and right from its orientation axis. If both motors are activated with the same positive or negative speed, the robot moves forward or backward. Differences between the motors' speeds lead to a rotation as well as a possible movement. The Braitenberg architecture provides direct connection between proximity sensors and motor actuators, thus forming a *reactive* control mechanism.

In addition, this paper applies the numerical potential field method [12] to make robots reach a target when the target`s location is known. The numerical potential field method constructs artificial potential fields incrementally by a wavefront expansion procedure. The potential values increase from the goal (lowest potential value) to all accessible positions in the environment. Then the robot follows the negated gradient to reach the goal.

There are three perception submodules in each robot.

In the first submodule, a camera on each robot provides the vision function just like the techniques in the literature [10] [16]. Objects with specific colors are regarded as intruders or emergencies. Robots can detect these objects by our color detecting algorithm.

In the second submodule, sixteen proximity sensors are arranged at the periphery of the robot`s body. These sensors` orientations are perpendicular to the robot periphery, thus they are oriented outwards. These sensors detect collision between the sensory rays and obstacles in the environment.

In the last submodule, each robot receives its geographic coordinates by appropriate devices, such as a global position system (GPS) or other self-localized systems. Therefore, each robot precisely knows its position.

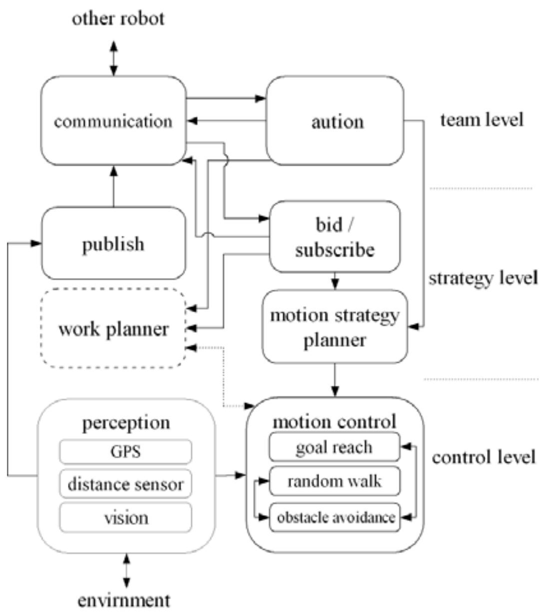

Figure 1 Robotic agent architecture

2) Strategy layer: This layer contains many individual modules, such as publishing tasks, bidding, and planning motion. The paper does not discuss the detail of work planner, therefore the work planner shows a dotted border in Fig 1. This layer makes longer-term decisions, involving the decision making of the task bidding and the calculation of the task allocation. The competitive bidding module receives the information from the team layer and computes the formation of the robot team. The bidding module also takes the time and the number of robots into consideration. The motion strategy planner determines the navigation of the robot and gives commands to the control layer for low level control.

3) Team layer: This layer deals with teamwork strategies, such as getting information from candidate robots and organizing the most efficient team. The communication module in this layer broadcasts and receives data for the auction bidding, and also manages both robot and task information. The structure and organization of robot teams are also stored here. This layer interacts with the robot strategy layer for further processing.

One argument is that an increase of the amount of transmitted data does not imply better performance on a collaborative multi-robot system. Hence, both in experiments and simulations, designing a suitable solution for the interpretation and usage of the transmitted information through communication is a challenge and a good research direction.

4.2 Auction Based Coordination Mechanism

Our coordination approach is inspired from [13] and [8]. A robot asks for help when finding a task by an auction mechanism. Each robot changes its role under different conditions. The robot becomes an auctioneer if it finds a task, or it becomes a subscriber to bid for the task.

The state transition diagram is shown in Fig. 2. Initially, robots patrol and search the area for tasks. If a robot finds a task, it informs the others, gives an auction to assign the best team for this task, and supervises them until the task is finished. On the other hand, if a robot in *search* state receives an auction signal, it bids for this task. If the robot wins the bid, it goes to the target, otherwise it returns to *search* state.

Each robot in a specific state performs the corresponding tasks and the details are as follows:

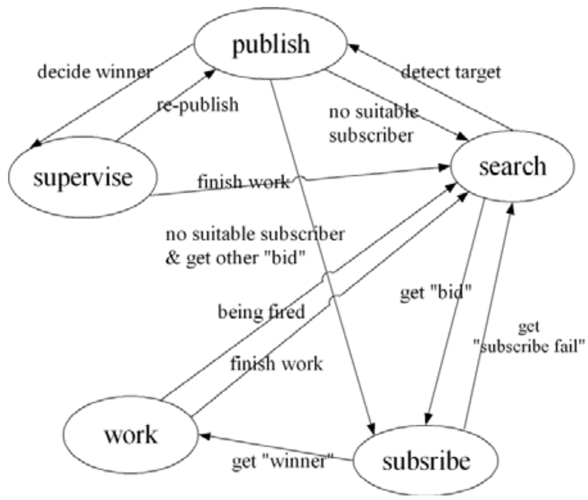

Figure 2 State transition diagram

1) Search: In the *search* state, robots walk randomly and search for targets by using a camera. When a target is detected, the robot changes to the *publish* state.

One argument is whether the random walk approach in *search* state is more efficient in detecting an emergency than just making robot sit and spin.

The main disadvantage of the random walk approach in *search* state is that the reliance on chance to find objects, intruders or emergencies [18].

Other approaches exist. In [9], the robots move forward initially, then turn left or right through some random arc after random times. In [1], a sit-and-spin behavior that enables robots to make a series of turns is used as the “wander-for-trash” behavior.

2) Publish: In the *publish* state, two tasks are undertaken. First, a robot in the *publish* state broadcasts the message about the target to the other robots. This paper assumes that the communication range is unrestricted. The message contains the information related to the detected target, such as name, needed capabilities and coordinations. Second, each robot computes its *fitness* value and sends the value back to the *publish* robot. Note that the *fitness* value is defined in Section III. During this period of time, the *publish* robot collects and compares all the *fitness* values to choose the optimal result that best fits the selected strategy. For example, under the *Best Efficiency Strategy*, if there is more than one robot team that satisfies capability requirement, the *publish* robot chooses the team which is able to accomplish the task fastest. After the *publish* robot determines the optimal team, it sends the result to each robot and changes to the *supervise* state.

3) Supervise: The *supervise* robot collects the "arrival" messages from all members of the winning team. The *supervise* robot gives a "start working" command to all members of the winning team after all members arrive. Afterward, the *supervise* robot takes the monitor role to watch the task progress. If the task is finished, the *supervise* robot and these *work* robots change all their states to the *search* states. There is a special condition for the *supervise* robot. If there exists members which cannot arrive in time at the target to perform the task, the *supervise* robot confirms the situation and sends a "fire" message to the absent robot and records this absence in memory. Remember, this paper assumes that the communication is unlimited.

4) Subscribe: When a *search* robot receives the message about a target from the *publish* robot, the *search* robot changes its state to *subscribe*. To bid the task, the *subscribe* robot computes the fitness value and sends the value back to the *publish* robot. If the *subscribe* robot receives a "winner" message, it goes to the target and performs the task.

5) Work: In this state, the *work* robot informs the *publish* robot when it arrives at the target. The *work* robot waits for all other teammates to arrive in order to perform the task together so that the winning team can tackle *tightly coupled* tasks. This paper assumes that, if all members of the winning team arrive at the target, the robots complete the task. This paper does not tell how robots tackle the task. Finally, when finishing the task, the *work* robots change their states to *search* states.

5. Experiment

This section describes the experimental results in the simulation. Three scenarios are conducted to evaluate the effectiveness of our approach. All the experiments are performed in the Webots system [20]. The following are the descriptions of three scenarios and of their primary results.

5.1 Deadlock

1) General Description: When multiple robots detect more than two targets at the same time and the robots cannot tackle these tasks simultaneously, a deadlock occurs.

2) *Simulation Scene:* Fig. 3 is the initial condition. Table 1 shows the robots` capabilities and the required resources of tasks. Both two tasks require two robots to accomplish. Each robot detects a task at the beginning, a deadlock occurs.

3) Simulation Result: This experiment uses a *timeout* mechanism to deal with the deadlock problem. Thus if the auctioneer fails for some time to receive an acknowledge after sending a new task message, the auctioneer checks the packages sent by other robots. If another task is published, the auctioneer gives up its *publish* state and changes to another state. In Fig. 4, the blue robot gives up its auctioneer role and cooperates with the green robot to deal with the red target.

Table 1 Specifications of deadlock scenario

number of robot	2
number of task	2
capabilities of robot1	AC
capabilities of robot2	BD
required resources of task1	ABC
required resources of task2	BCD

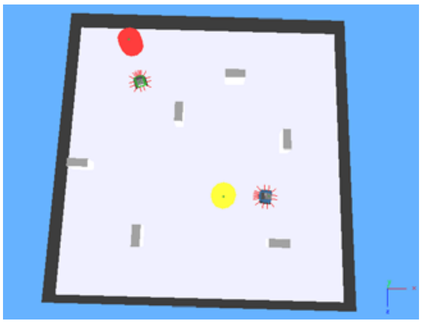

Figure 3 Initial condition of deadlock scenario

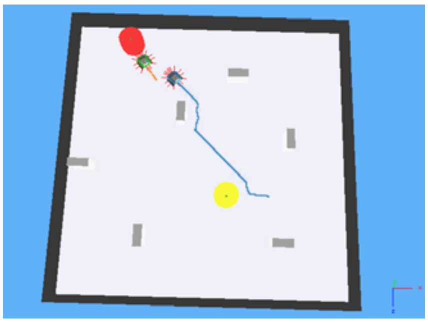

Figure 4 Final result of deadlock scenario

5.2 Specific Uncertainty

1) General Description: Robots might have difficulty in reaching the target in time (the robot happens to be broken, too many obstacles are in its way, etc.). If the auctioneer fails for some time to receive an "arrive" message from assigned robot, the auctioneer fires the assigned robot. Then, the auctioneer reassigns the task and gives a specific tag to the broken/blocked robot in order to keep track of its absence.

2) Simulation Scene: Fig. 5 shows the initial condition. The robot surrounded by obstacles denotes a broken one. The broken robot is closer to the task thus the robot is assigned the task at the beginning. Table 2 shows the robots` capabilities and the required resources of tasks.

3) Simulation Result: In Fig. 6, the auctioneer gives up the broken/blocked robot and reassigns the auction. Therefore, a robot farther away replaces the broken/blocked robot at the task.

Table 2 Specifications of uncertainty scenario

number of robot	4
number of task	1
capabilities of robot1 (auctioneer)	A
capabilities of robot2	B
capabilities of robot3	C
capabilities of robot4 (broken robot)	C
required resources of task	ABC

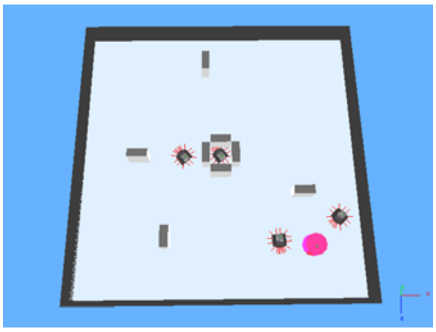

Figure 5 Initial condition of uncertainty scenario

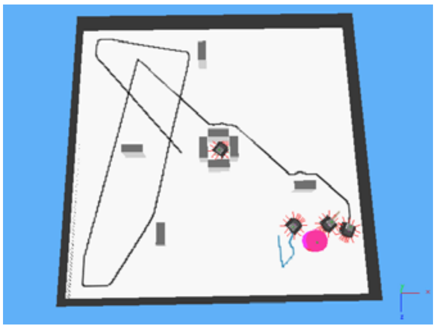

Figure 6 Final result of uncertainty scenario

5.3 Fewest Robot Strategy vs. Best Efficiency Strategy

1) General Description: This experiment compares two coordination strategies under the same conditions. This scenario assumes that the task can be done either by one robot or by multiple robots owing to the fact that either have the required capabilities. *Fewest Robot Strategy* (FRS) chooses the fewest number of robots as the optimal policy. The task completion time is not optimal but the strategy remains more robots to patrol the environment. *Best Efficiency Strategy* (BES) chooses the most efficient robot team so that the task completion time is minimized. The most efficient robot team means the team which arrives at task fastest.

2) Simulation Scene: Table 3 shows the robots` capabilities and the required resources of tasks. 100 tasks occur sequentially in particular period. This experiment tests five task occurrence periods: 2000, 3000, 4000, 5000, and 6000 cycles. A cycle is the duration of Webots simulation step. The duration value is assumed 64 milliseconds. If no suitable robot team are assigned to a task, the task disappears after the expiration time limit. This experiment uses two expiration time limits: 2000 and 5000 cycles.

Table 3 Specifications of comparison scenario

number of robot	4
number of task	100
capabilities of robot1 (auctioneer)	A
capabilities of robot2	B
capabilities of robot3	C
capabilities of robot4	ABC
required resources of each task	ABC

3) Simulation Result: Table 4 shows that BES can complete tasks more quickly both in two expiration time limits. Table 4 also shows that, the shorter the expiration time limit, the shorter the average task completion time. That is, the faster the successful tasks are accomplished. Tasks with shorter expiration time limit are prone to fail. Therefore only tasks, that are found easily and are assigned faster robot teams, become successful. As a result, the average task completion time becomes shorter.

Table 4 Average completion time results

Expiration time limit (unit:64ms)	BES	FRS
2000	2766.934	3011.138
5000	3233.445	3686.037

With expiration time limit 2000 cycle, Fig. 7 compares the average task completion time of two strategies in five task occurrence periods. Fig. 8 is the similar simulation with expiration time limit 5000 cycle. Fig. 7 and Fig. 8 both show that the longer the task occurrence period is, the shorter the average completion time is. This is because robots have sufficient time to accomplish tasks when the tasks occur less frequently. If tasks occur frequently, deadlock might happen. Thus robots need more time to deal with these tasks. Another explicit result is that BES has shorter average completion time than FRS.

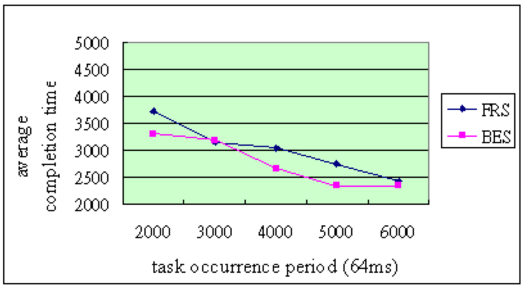

Figure 7 Comparison of FRS and BES for average task completion time under the expiration time limit 2000 cycles. The horizontal axis represents period of task occurrence and the vertical axis represents the average task completion time.

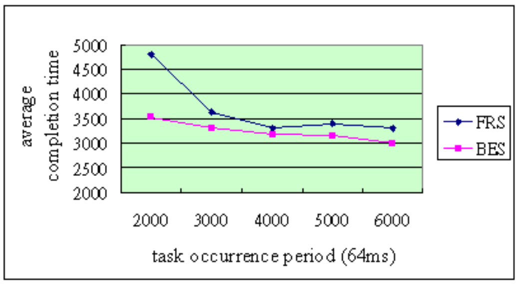

Figure 8 Comparison of FRS and BES for average task completion time under the expiration time limit 5000 cycles. The horizontal axis represents period of task occurrence and the vertical axis represents the average task completion time.

Table 5 shows the task success rate of BES is a little higher than that of FRS both in two expiration time limits. Fig. 9 and Fig. 10 provide the related information for the result of Table 5.

With expiration time limit 2000 cycle, Fig. 9 compares the task success rate of two strategies in five task occurrence periods. Fig. 10 is the similar simulation with expiration time limit 5000 cycle. The first result is that the longer the task occurrence period is, the higher the task success rate is. This is because robots have more chances to form an effective team to accomplish tasks in longer task occurrence period.

Both for BES and FRS, the longer the task occurrence period is, the higher the task success rate is (Fig. 9 and Fig. 10). This is mainly because, as the task occurrence period is greater than the average task completion time, the winning team has sufficient time to reach the target so that the winning team can tackle more discovered tasks. Thus the curves of Fig. 9 and Fig. 10 go high with task occurrence period.

The curves of FRS change more in degree. This is because, not only the task success rate highly relies on the task completion time, but we presume that the task discovery rate plays a significant role, especially in FRS. FRS assigns tasks to robot4 (team of fewest number of robots) no matter which robot is the auctioneer. Thus, while robot4 tackles the tasks, other robots keep patrolling the environment. Since FRS chooses the team of fewest number of robots to tackle the discovered tasks, there are more robots patrolling the environment compared to other strategies, which results in an increase in task discovery rate. Thus we presume that not only the short task completion time, but also the high task discovery rate may indirectly lead to the cause of high task success rate of FRS.

To what degree does the task discovery rate influence the task success rate? We need more experiments to examine this problem in the future.

Table 5 Task success rate results

Expiration time limit (unit:64ms)	BES	FRS
2000	74 %	70.8 %
5000	89.4 %	89.2 %

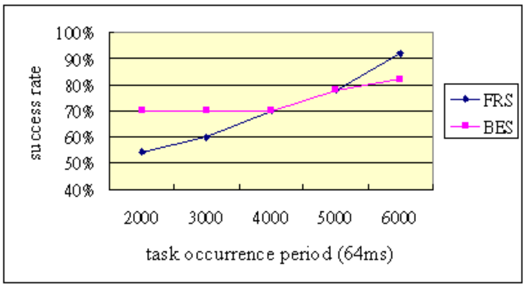

Figure 9 Comparison of FRS and BES for task success rate under the expiration time limit 2000 cycles. The horizontal axis represents period of task occurrence and the vertical axis represents the task success rate.

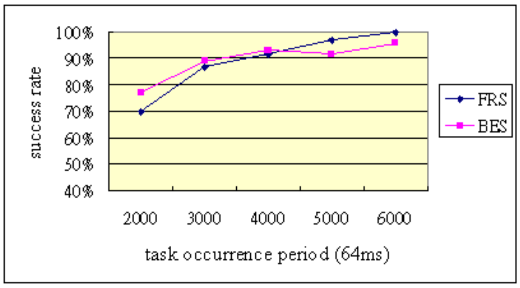

Figure 10 Comparison of FRS and BES for task success rate under the expiration time limit 5000 cycles. The horizontal axis represents period of task occurrence and the vertical axis represents the task success rate.

6. Conclusion

In the experiment, Table 4 and Table 5 show the efficiency of BES, but the success rate of BES is a little better than that of FRS. More experiments need to confirm this condition in the future.

In Section V-B, the *supervise* robot records the absent robot and keeps track of its absence. We plan to study more about the impact of this mechanism in the future. An interesting issue is that if the robots have the knowledge or information to predict the expiration time of the discovered tasks, some advanced algorithms need to be adopted.

In addition, this work is preliminary and we expect to implement future work by physical experiments. Formalizing capability representation is also another possible next step.

One of the disadvantages about market-based approaches is that the system heavily relies on reliable communication. This implementation also strongly assumes perfect communication. Therefore, we need to extend our research to the communication constraints.

In the future, developing more advanced strategies and comparing with other coordination methods for improving our understanding of multi-robot coordination are interesting and significant.

References

[1] Balch, T., Boone, G., Collins, T., Forbes, H., Mackenzie, D., and Santamar, J. C. "Io, ganymede, and callisto a multiagent robot trash-collecting team," *AI Magazine 16*, 2, 1995, pp. 39-51.

[2] Batalin, M. A., and Sukhatme, G. S. "Coverage, exploration and deployment by a mobile robot and communication network, " *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks 26*, 2, 2004, pp. 181-196.

[3] Braitenberg, V. *Vehicles: Experiments in synthetic psychlogy*. MIT Press, 1984.

[4] Dias, M. B., Zlot, R., Kalra, N., and Stentz, "A Market-based multirobot coordination: a survey and analysis," Tech. Rep. CMU-RITR-05-13, Carnegie Mellon University, 2005.

[5] Frias-Martinez, V., Sklar, E., and Parsons, S. "Exploring auction mechanisms for role assignment in teams of autonomous robots," In *Proceedings of he RoboCup Symposium*, 2004.

[6] Gage, D. W. "Command control for many-robot systems," In *Proceedings of AUVS-92, 19th Annual Technical Symposium and Exhibition of the Association for Unmanned Vehicle Systems, Huntsville AL*, 1992, pp. 22–24. Reprinted in Unmanned Systems Magazine, Fall 1992, volume 10, number 4, pp. 28-34.

[7] Gerkey, B., and Mataric, M. " Are (explicit) multi-robot coordination and multi-agent coordination really so different?" In *Proceedings of the AAAI Spring Symposium on Bridging the Multi-Agent and Multi-Robotic Research Gap*, Palo Alto, California, March 2004, pp. 1-3.

[8] Gerkey, B. P., and Mataric, M. J. " Sold!: Auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation 18*, 5, 2002, 758-768.

[9] Goldberg, D., and Mataric, M. J. " Robust behavior-based control for distributed multi-robot collection tasks, " Tech. Rep. IRIS-00-387, USC Institute for Robotics and Intelligent Systems, 2000.

[10] Jung, B., and Sukhatme, G. S. "Tracking targets using multiple robots: the effect of environment occlusion," *Autonomous robots 13*, 3, 2002, 191-205.

[11] Labella, T. H., Dorigo, M., and Deneubourg, J.-L. " Self-organised task allocation in a group of robots, " In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS04)*,Toulouse, France, June 23–25, 2004, R. Alami, Ed.

[12] Latombe, J.-C. *Robot motion planning*. Kluwer Academic Publishers, 1991.

[13] Lin, F.-C., and Hsu, J. Y.-J. " Cooperation protocols in multi-agent robotic systems, " *Autonomous Robots 4*, 2, 1997, 175-198.

[14] Lin, L., and Zheng, Z. "Combinatorial bids based multi-robot task allocation method, " In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)* 2005, pp. 1157–1162.

[15] Makarenko, A., and Durrant-Whyte, H. "Decentralized data fusion and control in active sensor networks," In *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm, Sweden, June 28-July 1 2004, pp. 479-486.

[16] Parker, L. E. "Distributed algorithms for multi-robot observation of multiple moving targets, " *Autonomous Robots 12*, 3, 2002, 231–255.

[17] Qi, H., Iyengar, S. S., and Chakrabarty, K. "Distributed sensor networks - a review of recent research," *Journal of the Franklin Institute 2001* , pp. 655-668.

[18] Rosencrantz, M., Gordon, G. J., and Thrun, S. "Locating moving entities in indoor environments with teams of mobile robots," In *Proceedings of 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS' 03)*, 2003, pp. 233-240.

[19] Tobe, Y., and Suzuki, T. "Wiser: Cooperative sensing using mobile robots," In *Proceedings of The First International Workshop on Heterogeneous Wireless Sensor Networks (HWISE-2005)*, 2005.

[20] WEBOTS. http://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.