Strategy-planned Q-learning Approach for Multi-robot Task Allocation

H. Hilal Ezercan Kayır and Osman Parlaktuna
Electrical and Electronics Engineering Department, Engineering and Architecture Faculty, Eskişehir Osmangazi University, Eskişehir, Turkey

Keywords: Multi-robot Task Allocation, Q-learning, Multi-agent Q-learning, Strategy-planned Distributed Q-learning.

Abstract: In market-based task allocation mechanism, a robot bids for the announced task if it has the ability to perform the task and is not busy with another task. Sometimes a high-priority task may not be performed because all the robots are occupied with low-priority tasks. If the robots have an expectation about future task sequence based-on their past experiences, they may not bid for the low-priority tasks and wait for the high-priority tasks. In this study, a Q-learning-based approach is proposed to estimate the time-interval between high-priority tasks in a multi-robot multi-type task allocation problem. Depending on this estimate, robots decide to bid for a low-priority task or wait for a high-priority task. Application of traditional Q-learning for multi-robot systems is problematic due to non-stationary nature of working environment. In this paper, a new approach, Strategy-Planned Distributed Q-Learning algorithm which combines the advantages of centralized and distributed Q-learning approaches in literature is proposed. The effectiveness of the proposed algorithm is demonstrated by simulations on task allocation problem in a heterogeneous multi-robot system.

1 INTRODUCTION

In most real-life robotic applications, multi-robot systems (MRS) are preferred instead of single robots to get desired system performance. The missions that cannot be accomplished by a single robot can easily be executed by a group of robots. Advantages such as faster task execution and robust system architecture are some of the reasons why MRS is frequently used in complex environments.

The major drawback of MRS is coordination requirements. Efficient system performance is directly related to accurate and precise coordination. Working environment of an MRS is generally dynamic and partially observable in nature. When designing an MRS, estimation of all possible situations that robots can face with is not possible. To overcome the problems encountered during system execution, robots have to adopt themselves and adjust their behaviours to changing environmental conditions. Consequently, the robots which have learning ability may play an important role to provide required group coordination and results in a more reliable system performance. The MRS with learning robots are more robust systems against the uncertainities and problems (Mataric, 1997).

In this study, a learning-based multi-robot task allocation approach is proposed to increase the system performance. If the robots carry their past task allocation experiences to a helpful knowledge that can be used for future task allocation process, it may be possible to enhance the system performance. For this purpose Q-learning algorithm is used. Q-learning algorithm is mainly used in environments that satisfy Markov Decision Process (MDP) properties (Yang ve Gu, 2004). Whereas, most MRS environments are not MDP. So, in this study, a new approach, Strategy-Planned Distributed Q-learning algorithm, is proposed to overcome the problems that appear in the application of Q-learning to multi robot systems.

The paper is organized as follows: In Section 2, a brief description of market-based multi-robot task allocation problem is given. In Section 3, the theory of Q-learning algorithm for both single-agent and multi-agent cases is presented. The proposed algorithms, learning-based multi-robot task allocation and Strategy-Planned Distributed Q-learning algorithms, are given in detail in Section 4

Hilal Ezercan Kayir H. and Parlaktuna O..
Strategy-planned Q-learning Approach for Multi-robot Task Allocation.
DOI: 10.5220/0005052504100416
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 410-416
ISBN: 978-989-758-040-6

and 5, respectively. Section 6 explains the experimental environment. In The experimental results are given in Section 7. Lastly, conclusion is given in Section 8.

2 MULTI-ROBOT TASK ALLOCATION

Multi-robot task allocation (MRTA) is defined as the problem of deciding which robot should perform which task in an appropriate order (Gerkey and Mataric, 2004). In multi-robot systems, robots and robot's task executing abilities are specified as system resources. In most cases, because of the scarcity of resources, it is not possible to complete all tasks (Jones et. al., 2007). This situation emphasizes the importance of efficient task allocation to get desired system performance. Task allocation process should be realized to provide an effective use of system resources by maximization of overall system gain or by minimization of system cost.

Market-based approaches present efficient solutions for coordination problems in multi-robot systems (Dias et al., 2006). In these approaches, each robot has its own decision-making mechanism but system coordination is realized by participation of all robots. Auction protocols are widely-used market-based task allocation methods (Gerkey and Mataric, 2002). Their major advantage is that they provide robust system architecture against system fault. In auction-based MRTA, task announcing unit acts as auctioneer and robots behaves as bidders. Tasks are also used as the items offered in auction process. The auction process is executed in the manner that the auctioneer announces the tasks, robots determine and send the bid values for tasks and the auctioneer decides the winner robot. In mobile robot applications, the bid values are calculated in terms of travelled distance, required time (Mosteo and Montano, 2007) or needed energy (Kaleci et al., 2010). The assignment of tasks to the robots is defined as the Optimal Assignment Problem (OAP) widely-used in operations research (Gerkey and Mataric, 2004). Hungarian Algorithm provides successful solutions for the OAP problem (Hatime, 2013). The auction process is completed when the winner robots are informed.

3 REINFORCEMENT LEARNING

Reinforcement learning methods are the machine learning approaches that do not require any input-output data sets or a supervisor (Russel and Norvig, 2003). In reinforcement learning, the agents are directly related to the environment by their perception and action units. The state transition of the environment results from the action of the agent. Agent is informed by a feedback signal called reward which indicates the effect of the action on the environment. Learning process is performed only through trial-and-error by using this reward value. No requirement of any supervisor, structural simplicity of the algorithms and the possibility of using it in partially observable and dynamic environments are the reasons why reinforcement learning is preferred especially in multi-agent system applications (Yang and Gu, 2004).

3.1 Single-agent Case

Markov Decision Proecess (MDP) is a tuple of $< S, A, P, \rho >$, where S is finite and discrete set of environment states, A is finite and discrete set of agent actions, $P: S \times A \times S \rightarrow \Pi(S): [0,1]$ is state transition function for each state-action pair and $\rho: S \times A \times S \rightarrow \mathbb{R}$ is reward function of the agent (Buşoniu et al., 2008). In standard reinforcement learning approaches, the environment is defined as an MDP.

For any discrete step k, the environment state changes from $s(k) \in S$ to $s(k+1) \in S$ by the action $a(k) \in A$ of the agent. The reward value of $r(k) = \rho(s(k), a(k), s(k+1))$, which the agent receives as the result of a(k), represents the instantaneous effect of action on the environment (Buşoniu et al., 2008). The agent determines which action should do through its action policy h. The agent in an MDP aims to maximize the expected value of the overall reward for each step. Action-value function is defined as the expected total gain of each action-state pair and it is given in equation (1).

$$\mathrm{Q^h(s,a)} = \mathrm{E}\left\{\sum_{\mathrm{i=0}}^{\infty} \gamma^{\mathrm{i}}\, \mathrm{r(k+i)} \middle| \mathrm{s(k)=s}, a(\mathrm{k})=a, \mathrm{h}\right\} \quad (1)$$

This equation is discounted sum of all future rewards where γ is the discount factor. Q-function is expressed as the optimal action-value function.

$$\mathrm{Q^*(s,a)} = \max_h \mathrm{Q^h}(s,a) \quad (2)$$

A learning agent should determine the optimal Q-

value, Q^* firstly and then should find required action by using action policy providing Q^*(Buşoniu et al., 2008).

Q-learning, is proposed by Watkins (Watkins, 1989), is a widely-used value function-based model-free reinforcement learning algorithm (Sutton and Barto, 1998). According to Q-learning algorithm, the optimal Q-value for each state-action pair is calculated by the following recursive equation:

$$Q(s(k),a(k)) = Q(s(k),a(k)) + \alpha\left[r(k) + \gamma \max_{a' \in A} Q_k(s(k+1),a') - Q(s(k),a(k))\right] \quad (3)$$

This equation does not require environment model and state-transition function. α is the learning rate and γ is the discount factor. If each state-action pair is repeated infinitely many and α is decreased in each step k, learned Q-values converges the optimal Q-values with the probability '*1*' (Watkins and Dayan, 1992).

3.2 Multi-agent Case

Stochastic Game (SG) is the extended form of MDP to multi-agent case. An SG is defined as the tuple of $< S, A, P, \rho_j >$, where S is the set of finite and discrete environment states, $A = A_1 \times A_2 \times \ldots \times A_m$ is the generalized action set for all agents, m is the number of agents, $P: S \times A \times S \rightarrow \Pi(S): [0,1]$ is the state transition function for each state-action pair and $\rho_j: S \times A \times S \rightarrow \mathbb{R}, j = 1 \ldots m$ is reward function for each agent (Buşoniu et al., 2008). For an SG, the state transitions are realized by joint actions of all agents.

One solution approach in an SG is to get the Nash equilibrium (Buşoniu et al., 2008). The Nash equilibrium is defined as the joint action policy such that each agent's action policy provides maximum total reward value against others' action policy (Yang and Gu, 2004). In the Nash equilibrium, it is not possible to increase the total reward by changing one agent's action policy while all other agents' action policies remain same.

Hu and Wellman are developed Nash-Q-learning algorithm which is based on reaching the Nash equilibrium (Hu and Wellman, 1998). It is shown that the optimal solutions are acquired under some certain conditions (Hu and Wellman, 2003). For each agent j, the Q-values are updated by equation (4).

$$\begin{aligned} V_N &= Nash_j(s, Q^1, \ldots, Q^j, \ldots, Q^m) \\ Q^j(s, a_1, \ldots, a_m) &= Q^j(s, a_1, \ldots, a_m) + \\ &\alpha[\rho_j + \gamma V_N - Q^j(s, a_1, \ldots, a_m)] \end{aligned} \quad (4)$$

If $\rho_1 = \cdots = \rho_m$ condition is valid, all agents aim to maximize the common goal and SG is called fully cooperative. In this case, the Nash equilibrium is expressed as follows.

$$Nash_j(s, Q^1, .., Q^j, .., Q^m) = \max_{a_1 \in A_1 \ldots a_m \in A_m} Q^j(s, a_1, .., a_m) \quad (5)$$

It is shown that a fully cooperative SG is assumed as an MDP (Boutlier, 1996). However, there exist more than one Nash equilibrium in an SG. It is very difficult problem to find joint actions which result in the common Nash equilibrium because all agents have independent decision making ability.

4 LEARNING-BASED MRTA

In a prior knowledge about the time sequence of tasks, it would be possible to optimize system performance by planning the order of tasks performed by each robot. But, such knowledge is not accessible for most multi-robot systems applications. Tasks appearing in an unpredictable time steps in random sequence affect the system performance in a negative manner. Especially, if there is a hierarchical order among the tasks in terms of priority or emergency, the tasks which must be completed primarily and unconditionally could not be done if the robots are busy with the low-priority tasks. As an example, consider a two-robot system R_1 and R_2, which perform low-priority tasks T_1 and T_2, respectively. If a high-priority task T_3 is announced before one of the robots finishes its task, T_3 will not be performed. This decreases the utility of the team, since T_3 is a high-priority task whereas T_1 and T_2 are low-priority tasks.

Generally in auction-based MRTA approaches, the robots bid for the announced tasks if they have the ability to do such a task and they are not busy at that time. These approaches have no mechanism for reasoning about future task sequence. It is clear that a precise estimation for future tasks is impossible. However, robots may have some expectations on the future task sequence when they use past experiences. For the above example, if one of the robots had a prediction about high-priority task sequence, this robot would not bid for the low-priority task and would wait for T_3 task.

Having such information about future task sequence is possible if robots have learning ability which transforms past experiences to a useful advice.

In this study, a learning-based task allocation approach is proposed to overcome the problem explained above. By the proposed method, robots

learn about the sequence relation between low-priority and high-priority tasks. The knowledge obtained in learning process is used to make decision about whether they attend the auction for an announced low-priority task or wait for a high-priority task.

5 STRATEGY-PLANNED DISTRIBUTED Q-LEARNING

Aim of the proposed approach in Section 4 is to enhance the system performance by means of past task allocation experiences. For this purpose, Q-learning algorithm is used.

The major difference between single-agent and multi-agent systems in terms of Q-learning is that for a single-agent the environment can be defined as MDP. However, in multi-agent case, the environment is no longer stationary because of the unpredictable changes which result from independent decision making and acting mechanisms of the agents. This is a contradiction to the essential assumption of MDP environment in Q-learning theory. Whereas the traditional Q-learning algorithm is successfully applied in single-agent systems, convergence to an optimal solution is not guaranteed in multi-agent case (Matignon et al., 2007).

In literature, there exist two fundamental approaches about the application of Q-learning in multi-agent systems. In this section, these two approaches are explained and a third approach, strategy-planned distributed Q-learning, which combines the advantages of these approaches, is proposed.

5.1 Centralized Q-Learning

In centralized Q-learning, the robots cooperatively learn common Q-values by considering joint actions. Q-values are updated by the following equation.

$$V = \max_{a'_1 \in A_1 \dots a'_m \in A_m} Q(s, a'_1, \dots, a'_m)$$
$$Q(s, a_1, \dots, a_m) = Q(s, a_1, \dots, a_m) + \alpha[\rho + \gamma V - Q(s, a_1, \dots, a_m)] \quad (6)$$

In this approach, learning process is realized either by each agent by observing all environmental changes or by a central unit communicating with all agents. It is noted that the environment is considered as MDP and optimal solutions can be converged because learning is executed using joint actions of all agents (Matignon et al., 2007). However, the dimension of the learning space which is defined as state space dimension times action space dimension becomes larger. For a fully cooperative SG with m agents, dimension of the learning space is given as follows:

$$|S||A_1| \dots |A_m| = |S||A|^m \quad (7)$$

Thus, the dimension of the learning space which increases exponentially depending on the number of agents results in huge computational load (Hu and Wellman, 2003).

Another disadvantage of the centralized Q-learning approach is the requirement of a central learning unit or explicit communication among robots (Wang and de Silva, 2006).

5.2 Distributed Q-Learning

Distributed Q-learning is the direct application of single-agent Q-learning to multi-agent case. In this approach, each individual agent learns its own Q-values as a result of its state and actions which is independent of other agents' actions. For each agent Q-values are updated by the following equation.

$$V = \max_{a'_j \in A_j} Q^j(s, a'_j)$$
$$Q^j(s, a_j) = Q^j(s, a_j) + \alpha[\rho_j + \gamma V - Q^j(s, a_j)] \quad (8)$$

The dimension of the learning space for each agent j is given as follows:

$$|S_j||A_j| \quad (9)$$

The advantages of the distributed Q-learning are small learning space and no requirement of inter-robot communication. The major disadvantage of distributed Q-learning approach is conflicting robot behaviours because of independent learning (Matignon et al., 2007).

5.3 Strategy-planned Distributed Q-Learning

In this study, to combine the advantages of centralized and distributed Q-learning approaches, a new Q-learning approach, Strategy-Planned Distributed Q-Learning method, is proposed. This method is also distributed in nature but it aims to remove behaviour conflicts of traditional distributed learning approaches. For this purpose, each cooperative robot is assigned a different learning strategy. For simplicity, the proposed algorithm is explained using a two- robot system as follows.

Definition 1. Let R_f and R_g be two robot s of a multi-robot system and Γ_f and Γ_g be the task sets of

these robots, respectively. If there exists a cooperative-task set such that

$$\Gamma_{coop} = \Gamma_f \cap \Gamma_g \; and \; \Gamma_{coop} \neq \emptyset \tag{10}$$

R_f and R_g are said cooperative robots.

Definition 2. For any task T_z, the base learning strategy, H^z_{base}, is defined as:

$$H^z_{base} \coloneqq \begin{cases} For \; T_z \; task \; type \\ select \; high - priority \; tasks \end{cases} \tag{11}$$

and complementary learning strategy is defined as:

$$H^z_{comp} \coloneqq \begin{cases} For \; T_z \; task \; type \\ select \; remaining \; low - priority \; tasks \end{cases} \tag{12}$$

Definition 3. If $|\Gamma_f| > |\Gamma_g|$ condition holds for $T_z \in \Gamma_{coop}$ task, the learning strategy of R_f and R_g robots are assigned as follows.

$$H^z_f \coloneqq H^z_{base} \tag{13}$$

$$H^z_g \coloneqq H^z_{comp} \tag{14}$$

The proposed approach has the advantage of small learning space because of its distributed structure. And also it prevents behaviour conflicts between agents through agents' different learning strategy unlike in traditional distributed learning approaches.

6 APPLICATION

To show the effectiveness of the proposed approach, an experimental environment on which the applications are realized is prepared. This environment has dimensions of 15x14 m, has eight rooms, corridors between rooms and a charging unit (Figure 1). The travelling paths between rooms are shown with dashed lines.

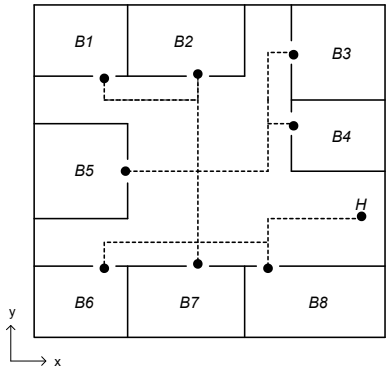


Figure 1: The map of experimental environment.

The multi-robot system used in applications consists of six robots, R_1, R_2, R_3, R_4, R_5, and R_6. The robots have different skills. Thus, the system is fully heterogeneous. There exist five different type of tasksT_1, T_2, T_3, T_4, and T_5. The robots and the tasks which the robots are capable of doing are given in Table 1.

Table 1: Tasks that can be performed by robots.

Robots	Tasks				
	T_1	T_2	T_3	T_4	T_5
R_1	+				
R_2	+	+			
R_3			+		
R_4			+	+	
R_5				+	+
R_6				+	

Tasks are generated randomly and with equal probability. The number of tasks announced at any time could be between two and five. Each task has two different priority degrees, low-priority and high-priority. Low-priority tasks and high-priority tasks are 65% and 35% of all tasks, respectively.

The applications are realized by using 45 experimental sets, each having nearly 50 tasks. First 30 sets are used for learning process and last 30 sets are used for test purpose. It is assumed that all the tasks allocated robots will be completed. None of the allocated tasks is left unfinished.

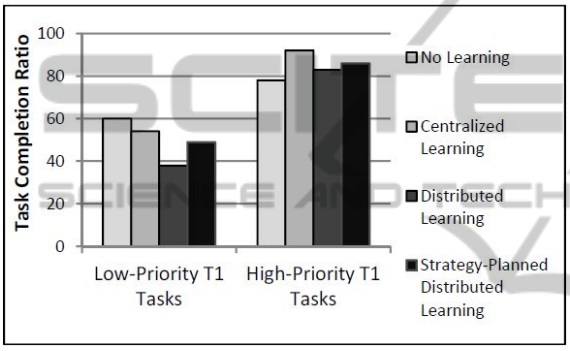
7 EXPERIMENTAL RESULTS

The purpose of the proposed approach is to increase system performance by using learning-based MRTA. The essential goal of the proposed approach is to increase the number of completed high-priority tasks. To show the effectiveness of the proposed approaches, the applications are realized in the experimental environment whose details are given in the previous section and the results are compared in terms of the task completion ratio. The task completion ratio is defined as the ratio of tasks assigned to any robot to the total number of tasks announced. Applications are executed for three learning approaches, centralized Q-learning, distributed Q-learning and strategy-planned distributed Q-learning, and the results are compared with the results when there is no learning. For each task type, the results for low-priority and high-priority tasks are given separately in Figure 2.

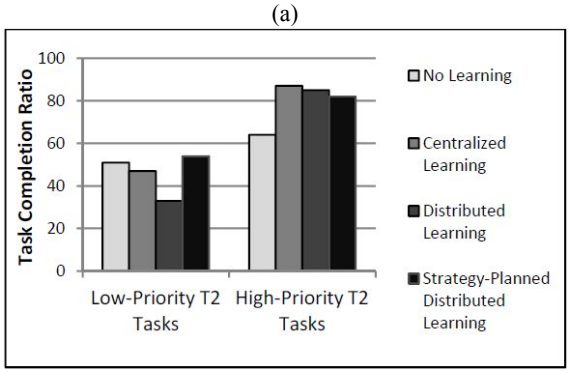
The graphs in Figure 2 show that the completion ratios of high-priority tasks for all task types are increased when any of the learning algorithms is used. This general result indicates that the robots are successfully benefited from the past task-allocation

experiences by their learning ability. However, the completion ratios of the low-priority tasks are decreased or remain nearly the same. In general, the number of total tasks to be completed is limited because of the limited system resources. Because of this, the results are acceptable.

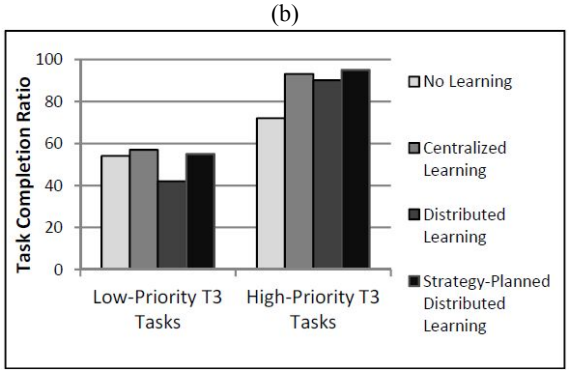
The centralized Q-learning approach is the learning algorithm that provides the highest task completion ratio for all high-priority and low-priority tasks because all joint actions of all robots are considered together in the learning process. Especially for the tasks that can be performed by more than one robot, while some of the robots wait for high-priority tasks, the others execute low-priority tasks.



(a)

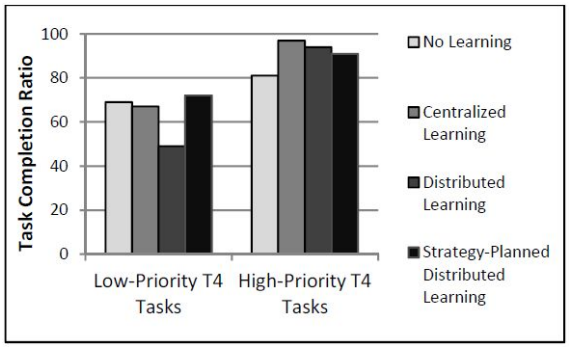


(b)

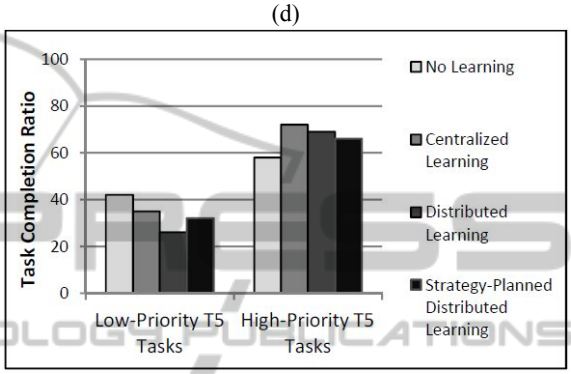


(c)

Figure 2: Task completion ratios of all task types a-e) T_1 to T_5.



(d)



(e)

Figure 2: Task completion ratios of all task types a-e) T_1 to T_5 (cont.).

In distributed Q-learning approach, each robot learns its state-action pairs without regarding the others' actions. Because all robots aim to perform high-priority tasks, task completion ratio gets higher for high-priority tasks whereas task completion ratio for low-priority tasks significantly decreases. The behaviour conflict that is the major disadvantage of distributed learning explains these results.

Results of the strategy-planned distributed Q-learning are not good as the results of centralized Q-learning but reasonably better than distributed Q-learning and no-learning cases. While task completion ratios of high-priority tasks for almost all task types are a bit less than the ones for other learning approaches, task completion ratios of low-priority tasks are significantly higher than that of the others.

As seen from the graphs, the best results are obtained in centralized Q-learning approach and the worst results are observed in distributed Q-learning as it is expected. From the point of view of learning space dimensions and computational load, the centralized approach has great disadvantage. For the system evaluated in this study, the dimension of learning space for distributed Q-learning and strategy-planned Q-learning is found as 102, whereas it is 1856 for centralized Q-learning. The

significant difference between these values is clear although the system considered here can be specified as a simple system. So, by taking into account the task completion ratios and computational loads, it is evident that the proposed approach, strategy-planned distributed Q-learning, yields appropriate and useful results.

8 CONCLUSIONS

In this paper, a new learning-based task allocation approach, Strategy-Planned Distributed Q-Learning, is proposed. Traditional Q-learning algorithm is defined in MDP environments. But MRS environments are no longer Markovian because of unpredicted behaviours of other robots and presence of uncertainties. There are two major approaches about Q-learning for multi-agent systems, distributed and centralized approaches. The proposed algorithm combines the advantages of distributed and centralized approaches. It is a distributed learning approach in nature but it assigns to robots different learning strategies in a centralized manner. Experimental results show that task completion ratio of high-priority tasks gets higher for all three learning approaches because the robots make use of their past task allocation experiences for future task execution through their learning ability. The experimental results show that the centralized learning approach produces the best solutions about task completion ratios of both high-priority and low-priority tasks. The proposed approach results in a bit less task completion ratios than centralized approach. However, it is indicated that the proposed algorithm provides reasonable solutions with its low learning space dimension and computational load.

REFERENCES

Boutlier C., 1996, Planning, learning and coordination in multiagent decision processes, *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '96, pp. 195-210.

Buşoniu L., Babuška R., Schutter B., 2008, A comprehensive survey of multiagent reinforcement learning, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.38, no.2, pp. 156-172.

Dias M. B., Zlot R. M., Kaltra N., Stentz A., 2006, Market-based multirobot coordination: a survey and analysis, *Proceedings of the IEEE*, vol. 94, no.7, pp. 1257-1270.

Gerkey B. P., Mataric M. J., 2002, Sold!: Auction methods for multi robot coordination, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758-768.

Gerkey B. P., Mataric M. J., 2004, A formal analysis and taxonomy of task allocation in multi-robot systems, *International Journal of Robotics Research*, 23(9), pp. 939-954.

Hatime H., Pendse R., Watkins J. M., 2013, A comparative study of task allocation strategies in multi-robot systems, *IEEE Sensors Journal*, vol. 13, no. 1, 253-262.

Hu J., Wellman M. P., 1998, Multiagent reinforcement learning: theoretical framework and an algorithm, *Proceedings of the Fifteenth International Conference on Machine Learning* ICML'98, pp. 242-250.

Hu J., Wellman M. P., 2003, Nash Q-learning for general sum games, *Journal of Machine Learning Research*, 4, pp. 1039-1069.

Jones E. G., Dias M. B., Stentz A., 2007, Learning-enhanced market-based task allocation for oversubscribed domains, *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, pp. 2308-2313.

Kaleci B., Parlaktuna O., Ozkan M., Kırlık G., 2010, Market-based task allocation by using assignment problem, *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 135-14.

Mataric M. J., 1997, Reinforcement learning in multi-robot domain, *Autonomous Robots*, 4(1), pp. 73-83.

Matignon L., Laurent G. J., Le Fort-Piat N., 2007, Hysteretic Q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams, *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, pp. 64-69.

Mosteo A. R., Montano L., 2007, Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation, *Proceedings of the IEEE International. Conference on Robotics and Automation*, pp. 3345-3350.

Russel S., Norvig P., 2003, *Artificial intelligence a modern approach*, Prentice Hall, New Jersey.

Sutton R. S., Barto A. G., 1998, *Reinforcement learning: an introduction*, MIT Press, Cambridge.

Wang Y., de Silva C. W., 2006, Multi-robot box-pushing: single-agent Q-learning vs. team Q-learning, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 3694–3699.

Watkins C. J., 1989, *Learning from delayed rewards*, University of Cambridge, UK, PhD Thesis.

Watkins C. J., Dayan P., 1992, *Q-learning*, Machine Learning, vol. 8.

Yang E., Gu D., 2004, Multiagent reinforcement learning for multi-robot systems: a survey, CSM-404, *Technical Reports of the Department of Computer Science, University of Essex*.