http://dx.doi.org/10.5755/j01.eie.25.2.23197

Q-Learning Based Failure Detection and Self-Recovery Algorithm for Multi-Robot Domains

Hatice Hilal Ezercan Kayir
Department of Electrical and Electronics Engineering, Engineering Faculty, Pamukkale University, Kinikli, Denizli, Turkey
hezercan@gmail.com

***Abstract*—Task allocation is the essential part of multi-robot coordination researches and it plays a significant role to achieve desired system performance. Uncertainties in multi-robot systems' working environment due to nature of them are the major hurdle for perfect coordination. When learning-based task allocation approaches are used, firstly robots learn about their working environment and then they benefit from their experiences in future task allocation process. These approaches provide useful solutions as long as environmental conditions remain unchanged. If permanent changes in environment characteristics or some failure in multi-robot system occur undesirably e.g. in disaster response which is a good example to represent such cases, the previously-learned information becomes invalid. At this point, the most important mission is to detect the failure and to recover the system initial learning state. For this purpose, Q-learning based failure detection and self-recovery algorithm is proposed in this study. According to this approach, multi-robot system checks whether these variations permanent, then recover the system to learning state if it is required. So, it provides dynamic task allocation procedure having great advantages against unforeseen situations. The experimental results verify that the proposed algorithm offer efficient solutions for multi-robot task allocation problem even in systemic failure cases.**

***Index Terms*—Autonomous systems; Intelligent robots; Multi-robot systems; Robot learning.**

I. Introduction

In recent years, multi-robot systems (MRS) have become more interested in a lot of areas varying from small indoor applications like home or office serving, museum guiding to more complex and sometimes dangerous fields such as search-and-rescue, fire fighting, underwater researches, mining, etc. The MRS provide concurrent processing and faster task execution features, distributed sensing and acting facilities and robust system architecture against the problems [1]. The key issue to benefit from these advantages and to reach desired system performance in MRS is that the multi-robot coordination should be done precisely and accurately [2]. In most real-life applications, all tasks cannot be accomplished because of the scarcity in the number of robots and their capabilities [3]. This reveals the effects and also necessity of efficient coordination mechanisms on system performance.

Multi-robot task allocation (MRTA) forms a basis for multi-robot coordination studies. MRTA is defined as the assignment of tasks to suitable robots in an appropriate order by aiming to optimize the system performance [4]. Auction protocol is one of the strategies given in the literature to solve MRTA problems [5]. In auction-based MRTA, tasks are simulated as items to be sold and they are announced to the robots which act as auctioneers. Each robot sends a bid representing the cost or profit of the task for its own. In mobile robot studies, bid values are generally calculated in terms of distance or time [6]. The winner robot is determined according to bid values by the way of maximizing utility or minimizing cost for overall system [7], [8]. So, system coordination is realized in a centralized manner, although each robot has its own decision-making mechanism. It is the major advantage of the auction-based strategies [9].

Most of the existing task allocation solutions are proposed for the applications which don't contain any uncertainty [10], whereas in real applications robots are faced with various difficulties to reach complete information about working environment because of various ambiguities [11]. In many cases, any information about in which order and how frequently that the tasks appear, cannot be accessible due to partially-observable and dynamic characteristics of working environment. Moreover, robots cannot predict the teammates' behaviour because each has independent decision-making mechanisms. This is why to make a perfect plan about system coordination is not possible [12]. To examine this problem, the kinds of uncertainties and their origins are investigated [10]. It proposes a task allocation approach based on interval data and applies for various levels of uncertainties in search-and-rescue tasks in disaster cases which are a good example of dynamic environments. It is claimed that on-line task allocation methods have much more successful results rather than off-line methods against to non-modelled characteristics of dynamic environments such as multi robot patrolling tasks [13]. Auction-based task allocation approaches are efficient way due to their dynamic structures [13], [14].

In the studies mentioned above, proposed approaches use instant decisions or actions of robots [13] or they require to model the uncertainties [10], [14]. But, this is not the case in real applications because of the nondeterministic features of environments especially in disaster areas [15], [16]. To ensure the optimized system coordination, it plays a significant role that robots adapt themselves to changing environmental conditions and rearrange their decisions and

Manuscript received 26 May, 2018; accepted 8 November, 2018.

actions. This becomes possible only if robots are equipped with learning abilities [12]. So, MRS provides adaptive and more reliable system architecture against the unpredictable situations [2]. A learning-based behaviour selection approach for noisy and dynamic MRS environment is studied and successful results are obtained [17]. An effective use of reinforcement learning for fire disaster response, which is a good example of dynamic task allocation problem, is examined in [18]. Reference [15] applies a learning-based approach for MRTA problems and tries to reason about future by task commitment in oversubscribed domains i.e. fire-fighting disaster. In another study, robots learn opportunity costs used as bid values for auction process in underwater exploration which is a kind of dynamic and unknown environments [19].

In most real-time MRS applications, tasks arise at unpredictable time steps during execution and the assignment of these tasks to robots is realized instantaneously. Especially in disaster-like environments, tasks must be done as quickly as possible although robots should clear a lot of hurdle firstly. These temporal and ordering constraints are explained as time-extended task allocation and it is added to Gerkey and Mataric's [4] classical MRTA taxonomy [16]. Similarly, to overcome time constraints task allocation is achieved by rescheduling procedure in time-extended manner [15], [20].

In this study, an auction-based instantaneous task allocation approach is used. According to this, tasks appear in a random sequence and at unpredictable time steps during execution. These tasks have to be immediately assigned to the robots. But this becomes possible only if robots are not busy with another task at that moment. When there is a hierarchical order among tasks, a crucial problem arises about achieving desired system performance. For example, a high-ordered task which is announced when all robots having capable of this task, execute another task, a low-ordered ones, cannot be assigned to any robot. To solve this problem, a learning-based MRTA approach similar to [3], [12] is executed. In this approach, robots use their past experiences for future task allocation process by learning to reasoning about task sequences. For this purpose, Q-learning, which is a widely used approach for MRS because it doesn't require environment model and easy to apply especially in dynamic environments [21], is preferred.

The used learning-based MRTA approach gives successful solutions to improve system performance unless a great change doesn't happen in environmental conditions. Additionally, it tolerates small environmental changes due to learning ability [3]. But in the case of failure in characteristics of the working environment or structure of MRS, the previously learned information becomes invalid. In a disaster case such as earthquake [10], fire-fighting [15], etc., great modifications occur in the sequence and ordering of the tasks [22], [23]. And also, a catastrophic failure of systems, i.e. some faulty robots may be out-of-order permanently, causes irretrievable decrease in system performance [13]. It is a major problem for real-time MRS applications that to detect such failure cases and to adapt robots' decision-making and acting mechanisms.

In this study, Q-learning based Failure Detection and Self-Recovery (FDSR) algorithm is proposed to overcome the problem mentioned above. According to the scenario designed as application environment, an extensive disruption in system characteristics during execution, i.e. changes in priority and ordering of tasks and their occurrence frequency, occurs. FDSR algorithm detects the failure cases and recovers the system to a reliable state which means that robots repeat the learning process according to new conditions. The novelty of this paper is that the proposed algorithm provides an adaptive task allocation procedure against dynamic system structure and also it ensures a great advance in system performance even in disaster cases by detecting the systemic or environmental failures.

The organization of paper is as follows: Section II gives brief information about Q-learning theory. In Section III, the problem examined in this study is stated. In Section IV, the proposed FDSR algorithm is presented. Application environment is presented, then experimental results and analysis is given in Section V. The paper ends with conclusion part in Section VI.

II. Q-LEARNING THEORY

Reinforcement learning (RL) is a machine learning approach which maps situations to actions by using reward signals. It does not need any supervisory information or any input-output relationship [24]. Environment transits to the next state as response to agent's current action and sends a reward signal to the agent. This reward signal represents how its action affects the environment. RL approaches are widely used in MRS applications because it works through trial-and-error concept with no system model requirement and it is relevant to use in dynamic environments [25].

A Markov Decision Process (MDP) is a sequential decision problem consisting of $\langle S, A, P, \rho \rangle$ where S is finite and discrete set of states, A is finite and discrete set of actions, $P: S \times A \times S \to \Pi(S) : [0,1]$ is probabilistic state transition function and $\rho : S \times A \times S \to \mathbb{R}$ is a real valued reward function. RL approaches are defined on the environments characterized as MDP [26].

In an MDP environment, agent's action $a_k \in A$ results in the change of state from $s_k \in S$ to $s_{k+1} \in S$ at any discrete time step k. Agent receives the reward value, $r_k = \rho(s_k, a_k, s_{k+1})$ as the measure of instantaneous effect of action a_k [24]. The aim of agent is to maximize the discounted sum of the expected reward at each step. The long-term total gain at step k, $Q^h(s, a)$ is given in (1) [21]

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

where $a_k = a, h$. Agent's action policy, h, is a function of state transition and $\gamma^i \in [0,1]$ is discount factor. Q function is the optimal action-value function and defined as in (2)

$$Q \times (s,a) = \max_h Q^h(s,a), \quad (2)$$

According to (2), agent obtains the optimal Q-value, Q^*, and then it specifies the action policy resulting in Q^* [27].

Q-learning algorithm is a RL approach that calculates the optimal Q-values for each state-action pair in an iterative manner as in (3) [29]

$$Q(s_k, a_k) = Q(s_k, a_k) + \\ +\alpha_k \left[r_k + \gamma \max_{a' \in A} Q(s_{k+1}, a') - Q(s_k, a_k) \right]. \tag{3}$$

In an MDP environment, the learned -Q-values converges to optimal Q^* values with probability ‘1’ as long as each state-action pair is repeated infinitely many times and learning rate α is diminished gradually in each step [29].

III. Problem Statement

In most MRS applications, tasks appear in a random sequence and unpredictable time steps during execution. This is the main reason that the planning about task ordering and sharing among the robots is not possible before system starts to work. Tasks can only be assigned to the robots that are not busy when they are announced. This means that some announced tasks cannot be executed if none of the robots are available. This situation causes that the desired system performance aren’t achieved especially when the tasks not performed have a priority such as emergency or sensitivity. As a solution for the mentioned problem, a learning-based task allocation method is proposed and successful result are obtained [12], [15].

The learning process of Q-learning algorithm is realized by repeating (3) infinitely many times for each state-action pair. However, in real applications, optimal Q^* values are reached in a finite iteration. For a state-action pair (s, a), the learned Q-value at iteration k is represented by $Q(s_k, a_k) = Q_k$. The normalized absolute error (NAE) value, $e_n(k)$, is defined as follows

$$e_n(k) = \left| \frac{Q_k - Q_{k-1}}{Q_k} \right|. \tag{4}$$

NAE value is “1” at the start of learning process and it gradually decreases. This means that the learned Q-values approximate to optimal Q^* values over enough iteration and NAE value gets close to zero. There exists such an iteration $k = k_L$ satisfying condition given in (5)

$$e_n(k) = \begin{cases} e_n(k) > \varepsilon_e, k < k_L, \\ e_n(k) < \varepsilon_e, k \geq k_L, \end{cases} \tag{5}$$

Threshold value ε_e has a very small value compared to “1”.

This condition can be thought as the stopping criteria of learning process. The learned Q-values are set as $Q^* = Q_{k_L}$.

In most Q-learning applications, the learning process, either offline or not, is stopped at the iteration k_L. The learned information is used later. This approach provides efficient solutions as long as working environment characteristics remain same [12]. In some cases, permanent variations such as change in number of tasks and occurrence frequencies or their priority levels may happen in the characteristics of environment during execution. Additionally, some robots may be out of order undesirably. Such a situation causes that the prior experiences of robots becomes invalid. It has great importance to detect these changes and to adapt the system to new conditions. For this purpose, Q-learning based Failure Detection and Self-Recovery (FDSR) algorithm is proposed in this study. FDSR algorithm detects the changes in environment, then reorganizes the MRS and restarts the learning process if these changes are permanent. So, it becomes possible to obtain a robust system against environmental changes. Detailed explanation of FDSR is given in the next section.

IV. Failure Detection and Self-Recovery Algorithm

(FDSR) algorithm assumes a heterogeneous MRS with m robots $(R_j, j = 1, \dots, m)$ having the ability to do n different types of tasks $(T_i, i = 1, \dots, n)$. Robots don’t have any knowledge about working environment at the beginning and each one learns for its own state-action pairs. FDSR algorithm proposes that the learning process goes on during execution, either active or passive; it continues to learn after the optimal Q^* values are reached. According to NAE values calculated at each step, robots choose one of three behaviours named as essential learning behaviour, hidden learning behaviour and failure detection behaviour.

A. Behaviour-1. Essential Learning Behaviour

Robots are in essential learning behaviour initially. This means that robots don’t have any knowledge about working environment yet. Learning process has just begun. Usual bidding strategy is valid such that a robot bids for tasks in its own task list unless is not busy for another task at that time. This behavior is active until the condition in (5) is met at iteration k_L where robots believe to be experienced enough. Optimal Q^* values are reached and it is set as $Q^* = Q_{k_L}$. At this point, essential learning behaviour ends and robot switches to hidden learning behaviour.

B. Behaviour-2. Hidden Learning Behaviour

In hidden learning behaviour, robots continue to calculate Q values and related NAE values although learning process is completed. So, optimal Q^* values are not updated so far. As long as the environmental characteristics remain same, Q-values are in a close neighbourhood of Q^* values and NAE is nearly zero. Robots in this behaviour bid in according to learned values when a task is announced.

If NAE value gets higher, robots notice that an unexpected variation occurs in characteristics of working environment. At iteration k_F that satisfies the condition in (6), robots think that something goes wrong. Then, robots transit to failure detection behaviour.

$$e_n(k) = \begin{cases} e_n(k) > \delta_F, k < k_F, \\ e_n(k) < \delta_F, k \geq k_F, \end{cases} \tag{6}$$

where $\varepsilon_e \ll \delta_F < 1$.

C. Behaviour-3. Failure Detection Behaviour

The aim of robots in failure detection behaviour is to specify status of changes in environmental conditions. In this behaviour, NAE value is determined by referencing Q^* values obtained at iteration k_L as shown in (7).

$$e_n(k)=\left|\frac{Q_k-Q^*}{Q_k}\right|. \tag{7}$$

where $k>k_F$. For consistency check of NAE values, $e_{av}(k)$ is defined as the arithmetic mean of NAE values calculated since failure as in (8)

$$e_{av}(k)=\frac{1}{k-(k_F-1)}\sum_{m=k_F}^{k} e_n(m). \tag{8}$$

This behaviour is a transient case and it takes (k_D-k_F) iterations at worst. If $e_{av}(k)$ decreases to a level less than δ_F before iteration k_D, robot realizes that everything was temporary and robots return to hidden learning behaviour.

If $e_{av}(k)$ is still higher than δ_F at iteration k_D, robot is sure a permanent changes in the system have occurred. Robots cancel the previous Q^* values; returns back to the essential learning behaviour and restart the learning process based on new environmental conditions.

V. APPLICATION AND RESULTS

The proposed algorithm is realized on a heterogeneous MRS with six robots (R_1, R_2, R_3, R_4, R_5, R_6,) capable of executing five different tasks (T_1, T_2, T_3, T_4, T_5). Each task has two priority degrees; low-priority and high-priority. This means that high priority tasks have high degree of importance, emergency or sensitivity. If a robot has ability to do a task, it can perform both low and high priority of that task. Robots and related tasks are shown in Table I by '✓' sign. Pioneer P3-DX robots' realistic models are used during experiments and all tasks are defined as in real-life.

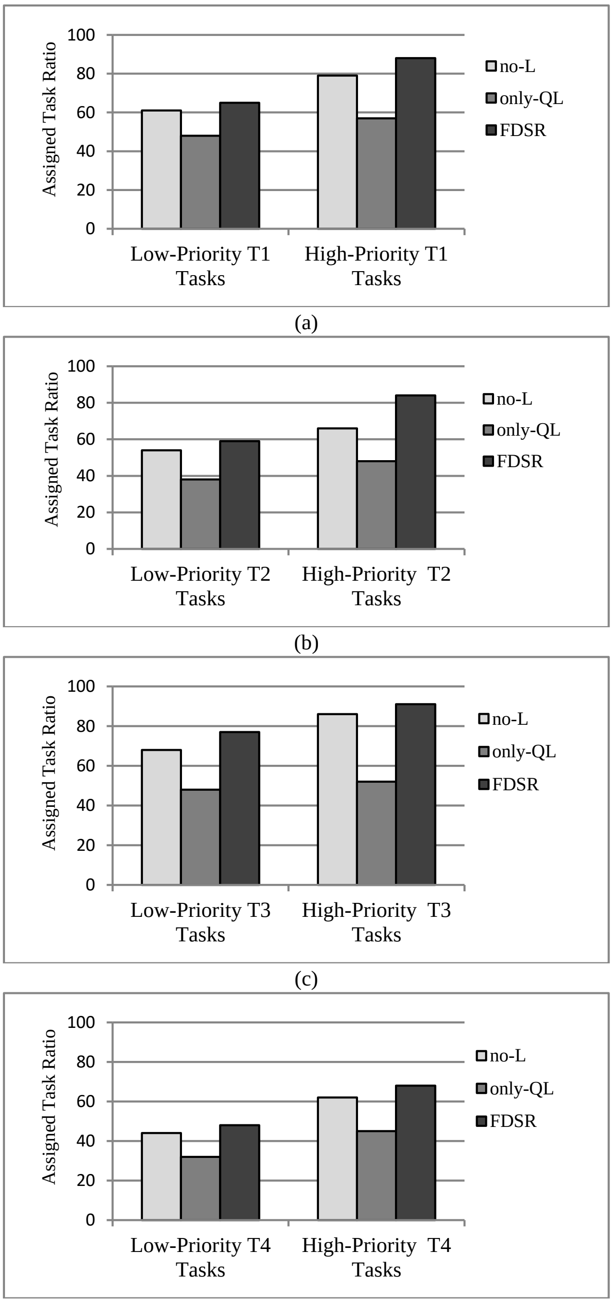
TABLE I. ROBOTS AND RELATED TASKS.

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

To represent the working environment, two different scenarios are defined. The first scenario represents the starting configuration and the second one exemplifies the environment after permanent changes occur. In the first scenario, all task types are equally probable and each one has low-priority and high priority tasks with ratio of 65 % and 35 % respectively. In the second scenario, the tasks don't occur with equal probability. The percentage of the tasks becomes 25 %, 20 %, 25 %, 20 % and 10 % respectively. In addition, the percentages of low-priority and high priority tasks becomes 55 % and 45 % for T_2 and 50 % and 50 % for T_3. At the beginning, the first scenario is valid and the second scenario becomes active at the one third of working duration.

The main purpose is to raise the number of completed high-priority tasks while keeping the total number of completed tasks as high as possible. Assigned Task Ratio (ATR) term is used as performance criteria for proposed algorithm. ATR is defined as the percentage of the number of assigned tasks to the number of all announced tasks. It is essential assumption that all tasks assigned to the robots are finished. To show the effectiveness of the proposed algorithm, experiments are realized for three methods named as no-L, only-QL and FDSR. The first method, no-L, represents the no learning case with usual bidding strategy. The other method, only-QL, uses a Q-learning based MRTA, similar to [12].

FDSR is the proposed approach in this study. The results of low-priority and high-priority tasks for each task are given separately in Fig. 1 for these three methods.

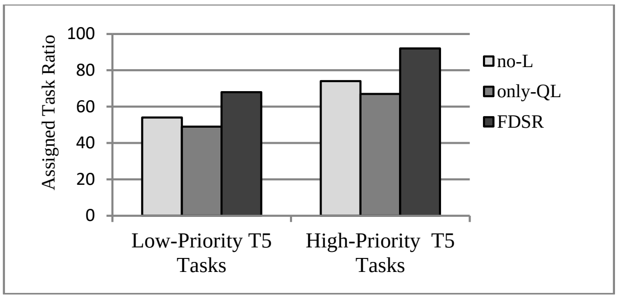


(a)

(b)

(c)

(d)



(e)

Fig. 1. Assigned Task Ratio of low-priority and high-priority tasks for each task type: a) T_1 tasks; b) T_2 tasks; c) T_3 tasks; d) T_4 tasks; and e) T_5 tasks.

It is seen from the graphs in Fig. 1 that ATR of high-priority tasks are almost higher than the low-priority tasks for all methods due to used auction strategy. only-QL method learns about working environment at the beginning of the system and then stops. Because the learned values are not suitable to the environment characteristics after failure and robots continue to obey their prior experience, ATR of all tasks get lower. This point out that inappropriate learning causes undesired results. FDSR method aims to find out the environmental changes and to specify whether these are permanent or not. If permanent, FDSR recovers the system to a reasonable start state, e.g. cancels the previously-learned values and restarts the learning process for new environmental conditions. The success and efficiency of FDSR algorithm can easily be observed from the graphs in Fig. 1. ATR values for low and high-priority task of all tasks are higher compared to other two methods.

VI. Conclusions

In this study, Q-learning based failure detection and self-recovery (FDSR) algorithm is proposed for task allocation problems in dynamic multi-robot domains. The aim of this algorithm is to detect the environmental changes and to recover the system to a reliable state when these changes are permanent as in the case of disaster. The proposed algorithm derives three behaviours as essential learning behaviour, hidden learning behaviour and failure detection behaviour. The results of FDSR algorithm are compared with the results of no-L and only-QL algorithms. Experimental results indicate that the algorithm provides efficient solutions to achieve desired system performance in terms of assigned task ratio when any permanent changes occur in environment characteristics undesirably.

References

[1] R. C. Arkin, *Behavior-Based Robotics*. Cambridge: MIT Press, 1998, pp. 358–420.

[2] M. J. Mataric, “Reinforcement learning in multi-robot domain”, *Autonomous Robots*, vol. 4, no. 1, pp. 73–83, 1997. DOI: 10.1023/A:1008819414322.

[3] H. H. Ezercan Kayir, “Experienced-task based multi robot task allocation”, *Anadolu University of Sciences & Technology – A: Applied Sciences & Engineering*, vol. 18, no. 4, pp. 864–875, 2017. DOI: 10.18038/aubtda.340101.

[4] B. P. Gerkey, M. J. Mataric, “A formal analysis and taxonomy of task allocation in multi-robot systems”, *Int. Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004. DOI: 10.1177/0278364904045564.

[5] B. P. Gerkey, M. J. Mataric, “Sold!: auction methods for multi robot coordination”, *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002. DOI: 10.1109/TRA.2002.803462.

[6] A. R. Mosteo, L. Montano, “Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation”, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Roma, Italy, 2007, pp. 3345–3350. DOI: 10.1007/978-3-642-04686-5_26.

[7] R. Zlot, A. Stentz, “Market-based multirobot coordination for complex tasks”, *Int. Journal of Robotics Research Special Issue on the 4th Int. Conf. on Field and Service Robotics*, vol. 25, no. 1, pp. 73–101, 2006. DOI: 10.1177/0278364906061160.

[8] H. Hanna, “Decentralized approach for multi-robot task allocation problem with uncertain task execution”, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Alberta, Canada, 2005, pp. 535–540. DOI: 10.1109/IROS.2005.1545037.

[9] M. B. Dias, R. M. Zlot, N. Kaltra, A. Stentz, “Market-based multirobot coordination: a survey and analysis”, in *Proc. IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006. DOI: 10.1109/JPROC.2006.876939.

[10] N. Hooshangi, A. A. Alesheikh, “Agent-based task allocation under uncertainties in disaster environments: an approach to interval uncertainty”, *Int. Journal of Disaster Risk Reduction*, vol. 24, pp. 160–171, 2017. DOI: 10.1016/j.ijdrr.2017.06.010.

[11] L. Liu, D. A. Shell, “Assessing optimal assignment under uncertainty: an interval based algorithm”, *Int. Journal of Robotics Research*, vol. 30, no. 7, pp. 936–953, 2011. DOI: 10.1177/0278364911404579.

[12] H. H. Ezercan Kayir, O. Parlaktuna, “Strategy planned Q-learning approach for multi-robot task allocation”, in *Proc. 11th Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO 2014)*, Vienna, Austria, 2014, vol. 2, pp. 410–416. DOI: 10.5220/0005052504100416.

[13] A. Farinelli, L. Iocchi, D. Nardi, “Distributed on-line dynamic task assignment for multi-robot patrolling”, *Auton. Robot.*, vol. 41, no. 6, pp. 1321–1345, 2017. DOI: 10.1007/s10514-016-9579-8.

[14] J. Tang, K. Zhu, H. Guo, C. Gong, S. Zhang, “Using auction-based task allocation scheme for simulation optimization of search and rescue in disaster relief”, *Simulation Modelling Practice and Theory*, no. 82, pp. 132–146, 2018. DOI: 10.1016/j.simpat.2017.12.014.

[15] E. G. Jones, M. B. Dias, A. Stentz, “Learning-enhanced market-based task allocation for oversubscribed domains”, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, San Diego, USA, 2007, pp. 2308–2313. DOI: 10.1109/IROS.2007.4399534.

[16] E. Nunes, M. Manner, H. Mitiche, M. Gini, “A taxonomy for task allocation problems with temporal and ordering constraints”, *Robotics and Automation Systems*, no. 90, pp. 55–70, 2017. DOI: 10.1016/j.robot.2016.10.008.

[17] E. Martinson, A. Stoytchev, R. C. Arkin, “Robot behavioral selection using Q-learning”, in *Proc. IEEE/RJS Int. Conf. of Intelligent Robots and Systems*, Lausanne, Switzerland, 2002, pp. 970–977. DOI: 10.1109/IRDS.2002.1041516.

[18] Y.-T. Tian, M. Yang, X.-Y. Qi, Y.-M. Yang, “Multi-robot task allocation for fire-disaster response based on reinforcement learning”, in *Proc. Eighth Int. Conf. on Machine Learning and Cybernetics*, Baoding, 2009, pp. 2312–2317. DOI: 10.1109/ICMLC.2009.5212216.

[19] J. Scheider, D. Apfelbaum, D. Bagnell, R. Simmons, “Learning opportunity costs in multi-robot market based planners”, in *Proc. Int. Conf. on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, 2005. DOI: 10.1109/ROBOT.2005.1570271.

[20] J. Turner, Q. Meng, G. Schaefer, A. Whitbrook, A. Soltoggio, “Distributed task rescheduling with time constraints for the optimization of total task allocations in a multirobot system”, *IEEE Trans. Cybernetics*, no. 99, pp. 1–15, 2017. DOI: 10.1109/TCYB.2017.2743164.

[21] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998, pp. 55–69.

[22] J. Tang, K. Zhu, H. Guo, S. Zhang, “Simulation optimization of search and rescue in disaster relief based on distributed auction mechanism”, *Algorithms*, vol. 10, no. 4, pp. 125–141, 2017. DOI: 10.3390/a10040125.

[23] M. Gini, “Multi-robot allocation of tasks with temporal and ordering constraints”, in *Proc. Thirty-first AAAI Conf. on Artificial Intelligence (AAAI 2017)*, San Francisco. California, USA, 2017, pp. 4863–4869. DOI: 10.1016/j.robot.2016.10.008.

[24] E. Yang, D. Gu, “Multiagent reinforcement learning for multi-robot systems: a survey”, Technical Reports of the Dept. of Computer Science, Univ. of Essex, 2004.

[25] L. Busoniu, L. Babuska, B. Schutter, “A comprehensive survey of multiagent reinforcement learning”, *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 38, no. 2, pp. 156–172, 2008. DOI: 10.1109/TSMCC.2007.913919.

[26] L. P. Kaelbling, M. L. Littman, A. W. Moore, “Reinforcement learning: a survey”, *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996. DOI: 10.1613/jair.301.

[27] C. J. Watkins, P. Dayan, “Q-learning”, *Machine Learning*, vol. 8, no. 3–4, pp. 279–292, 1992.