Market-Based Dynamic Task Allocation using Heuristically Accelerated Reinforcement Learning

José Angelo Gurzoni Jr., Flavio Tonidandel, and Reinaldo A. C. Bianchi

Department of Electrical Engineering
Centro Universitário da FEI, São Bernardo do Campo, Brazil
jgurzoni@ieee.org, {flaviot, rbianchi}@fei.edu.br

Abstract. This paper presents a Multi-Robot Task Allocation (MRTA) system, implemented on a RoboCup Small Size League team, where robots participate of auctions for the available roles, such as attacker or defender, and use Heuristically Accelerated Reinforcement Learning to evaluate their aptitude to perform these roles, given the situation of the team, in real-time.
The performance of the task allocation mechanism is evaluated and compared in different implementation variants, and results show that the proposed MRTA system significantly increases the team performance, when compared to pre-programmed team behavior algorithms.

Keywords: Multi-Robot Task Allocation, Reinforcement Learning, RoboCup Robot Soccer.

1 Introduction

Effective cooperation in teams of autonomous robots, operating in highly dynamic environments, poses a significant challenge. The robotic team members have to adapt or change their assigned tasks in real-time, in response to new and possibly unforeseen situations, while ensuring the team's long-term goals progression. Under these circumstances, efficient dynamic task allocation systems are desirable even in small and homogeneous teams, if only to minimize physical interference. However, on larger, often heterogeneous teams, it becomes essential.

Designing these allocation systems, often referred to as Multi-Robot Task Allocation (MRTA) systems, involves dealing with a straight-forward question, but of difficult answer ([8]): "which robot should execute which task?". To answer this question, robots need to perceive their environment, evaluate their aptitudes and communicate with their teammates, to avoid interferences, effort duplication, and deficient task resolution. Such challenges motivated researchers to develop a number of solutions to solve robotic task allocation problems. In-depth surveys of the MRTA field can be found on [13], [8], and [9].

Achieving fully cooperative behavior into a team of the RoboCup Small Size League (SSL) is specially challenging because, besides being also an adversarial domain, in an SSL game, the number of robots involved is relatively large and these robots are highly dynamical, able to reach speeds above five meters per second. Although many works addressed team play behavior and cooperation in robot soccer, most of them dealt with specific actions or situations, such as passing the ball [11]. Few addressed the creation of fully cooperative team architectures, such as the STP [2], [3]. The paper presents a task allocation system that can contribute towards the creation of a fully cooperative architecture, and that is flexible enough to be attached to an existing strategy system.

The proposed MRTA system uses auctions to allow an autonomous strategy expert, the Coach module, to offer roles for the robots to perform during the game. The robots use a reinforcement learning accelerated by heuristics [1] [4] technique, the HAQ(λ), to learn their fitness for each of the roles, given the present game situation.

The auctioning forms a computationally cheap and efficient method for achieving team behavior, while the reinforcement learning allows the robots to reason by themselves about their functions on the team.

The rest of this paper is organized as follows: Sec. 2 discusses the market-driven methods for task allocation in robotic teams, while Sec. 3 describes the heuristically accelerated reinforcement learning algorithm used. The proposed MRTA system is described in Sec. 4, and experiments and results are shown in Sec. 5. The paper ends with conclusions and future works, in Sec. 6.

2 Market-Based Methods for MRTA

Market-based methods take inspiration from the theory of market economies, where self-interested agents and groups trade goods and services, seeking to maximize their own profits, and while doing that inherently improve their economy as a whole, making it more efficient. These market-based methods are centered on the concept of *utility* functions (sometimes called cost or profit functions), which represent the ability of the agent to measure its own interest in a particular item available for trading. In MRTA systems, utility functions commonly express some measure of the robot's fitness towards performing a certain task, a function of the cost estimated to perform it or a junction of both.

Several market-based approaches were developed for multi-robot coordination, with different characteristics. A good survey on these approaches is provided on [6] and [9].

Many market-based MRTA systems operate, in one way or another, through auction mechanisms. In general, auctions are, scalable, computationally cheap, and have reduced communication requirements. They can be performed centrally, by an auctioneer, or by the robots themselves, in a distributed way. The MURDOCH [7] architecture, for example, uses first-price auctions, where robots submit their bids for the tasks being offered, and the highest value wins. This architecture also allows robots to renegotiate, selling their tasks to other robots,

and contracts are time-limited (the auctioneer can reclaim a task after some time), two features that give fault-tolerance capabilities to the architecture.

Another interesting architecture is the TraderBots [5], which employs a fully distributed, fault-tolerant, trading system and applies it into spatial exploration robotic teams. In the TraderBots, a more sophisticated economy is created, where robots seek to accumulate profits on the long-term, and can also subcontract as a mean to take profits for giving guidance to other robots. The MRTA system proposed in this paper takes a simpler approach, primarily because of the different goals in the robot soccer domain. Nevertheless, these architectures are rich references for those who want to understand market-based task allocation systems.

In a work similar to the proposed in this paper, Kose et al. [10] applied an market-based MRTA system with Q(λ) to simulated robot soccer, but in a different abstraction level. In that work, the MRTA system is applied directly to the control of the robot actions, such as to kick or to defend the goal, while in this paper the MRTA system operates on a higher level abstraction. This paper also proposes the use of heuristic functions into the reinforcement learning algorithm.

Finally, one interesting direction, not taken in this work, would be to explore combinatorial auctions and other dynamic programming techniques, as described in [14]. These methods may improve performance, as they do not act similarly to greedy task schedulers like first-price auctions do.

3 Heuristically Accelerated Q(λ)

In a Reinforcement Learning problem, the agent learns through interactions with the environment, by experience, without the need for an environment's model. On each interaction step, the agent senses the current state s, takes an action a, altering the state s, and receives a reinforcement signal r. The agent's goal is to learn a policy π that maximizes its returns. The Q(λ) algorithm is an extension of the popular Q-Learning [21], where reinforcements are not given only for the terminal states, but also to the states recently visited, making the convergence to the optimal policy faster. The Q(λ) update rule is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a) \tag{1}$$

$$\delta \leftarrow r + \gamma Q(s',a^*) - Q(s,a) \tag{2}$$

$$e(s,a) \leftarrow \gamma\lambda e(s,a) \tag{3}$$

Where s is the current state; a is the action performed in s; r is the reward received; s' is the new state; γ is the discount factor ($0 < \gamma < 1$); α is the learning rate; and λ controls the decay in the reinforcement for states farther in the past.

The action selection of the algorithm is:

$$a \leftarrow arg\max_{a}(Q(s',a)) \tag{4}$$

A heuristically accelerated reinforcement learning algorithm [1], including the Heuristically Accelerated Q(λ), is a way to solve the RL problem with explicit use of a heuristic function $H : S \times A \rightarrow \Re$ for influencing the choice of actions taken by the learning agent. The heuristic serves as a starting point to the agent, a prior knowledge about the domain that biases the decision towards one of the options. It helps the algorithm to converge faster. The heuristic $H_t(s, a)$ indicates the importance of performing action a when visiting state s, at time t. The use of heuristics does not alter the convergence proofs of the Q(λ) ([21], [17]), because the only change introduced is in the action selection, that is modified from (4) to:

$$a \leftarrow arg\max_{a} [Q(s', a) + \xi H(s, a)] \tag{5}$$

Heuristic functions are flexible, they can be adapted or modified on-line, as learning evolves and new information becomes available, and either prior domain information or knowledge acquired during initial stages of the learning can be used to define heuristics.

The next section shows the implemented task allocation system.

4 The Implemented Task Allocation System

The proposed MRTA system, shown in Fig. 1, has essentially two parts: (i) the auctioning module, that uses first-price auctions as task allocation mechanism, as shown in Sec. 2, and (ii) the RL module described in Sec. 3, which is present in each robot and uses the HAQ(λ) algorithm to learn the robot's utility functions, or interests, towards bidding for each of the roles offered on the auction. These modules are described in details along this section, after a brief outline of the strategy system where the MRTA is inserted in.

Market-based methods use the sound theory of market equilibrium, from economy. They are computationally efficient, and can be enhanced with machine learning algorithms in a simple manner, in the form of utility functions. However, the efficiency of market-based methods is only as good as these utility functions, used by the agent to reason about its aptitudes and interests. As the creation of high-level reasoning algorithms that could serve as utility functions in robot soccer is a difficult task, due to the complexity and dynamism of the environment, using reinforcement learning as the machine learning algorithm is attractive, as RL can learn by experiences, without environment models. Also, as shown in Sec. 3, the encoding of the domain's knowledge in the form of heuristics allows faster convergence of the RL algorithm.

The proposed MRTA system is not inspired only by market-based methods, though. As many of the more recent task allocation architectures [18], [12], it has characteristics from other MRTA paradigms, such as the use of roles, commonly found in socially inspired MRTA systems.

The remainder of this section briefly describes the strategy module where the task allocation is performed, to help on its understanding, and then presents a description of the actual MRTA system proposed.

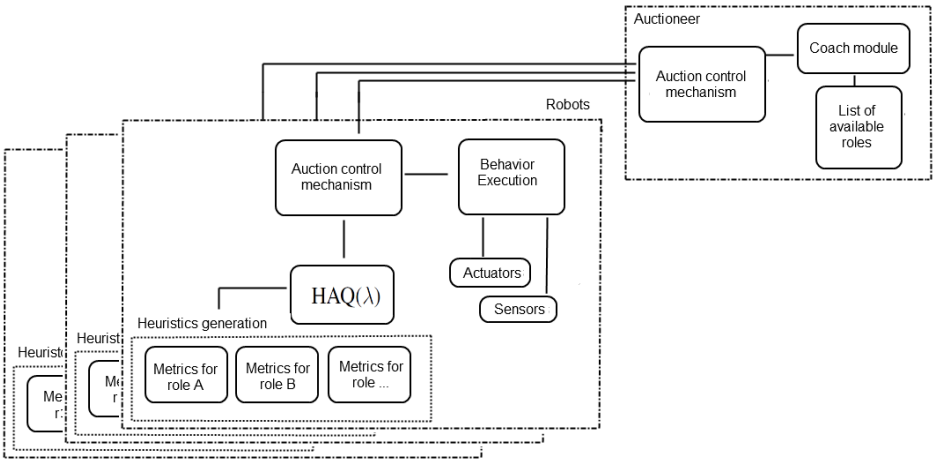

Fig. 1. Block diagram of the implemented MRTA system, showing the modules of the participating robots and the auctioneer.

4.1 Strategy System

This MRTA system is implemented in a strategy module formed by three abstraction layers: *primitives, skills* and *roles*. The lowest layer has the *primitives*, that are simple actions like activating the kicking or dribbling devices, or the ball presence sensor. On top of this layer is the *skills* layer, that contains short duration actions which involve the use of one or more primitives and additional computation, such as speed estimation, forecasting of the positions of objects and measurement of the completion of primitive tasks. Passing the ball or aiming and shooting to the goal are examples of skills.

The top layer is the *roles* layer, which are created using combinations of skills and the logic required to coordinate their execution, and are intended to be executed for longer periods. The existent roles are fullback, defender, midfielder, striker, forward and attacker.

4.2 Auctioning Module

The Auctioning module is executed by the Coach agent, responsible for analyzing the game situation and defining, according to its reasoning of the conditions, what roles, and in how many instances, will be available for the robots to bid. At a certain stage of the match, for instance, the Coach can decide that the team should be more offensive, and then auction more offensive instances, like Strikers and Attackers, and only one instance of the Defender role. A note: the number of roles offered can be larger than the number of robots, so as to give to the robots more selection choices. The Coach also prioritizes the order in which the roles will be offered, starting by the most offensive when the team is attacking and by the most defensive otherwise.

The first-price auction algorithm works as follows:

1. **Auction announcement.** The coach agent starts an auction, offering the role of highest priority available. A message is sent to the robots, informing about the open auction and the role being offered.
2. **Biddings formulation.** Each robot evaluates its utility function and submits a bidding towards that role.
3. **Auction result.** The coach defines the winner of the auctioned role and sends a message to the robots, informing them.
4. **Repetition.** The process is repeated, without the winner robot and the previously auctioned role, until there are no more robots without tasks.

An important parameter for the auctioning module is the interval between auctions. If the auctioneer could monitor the robot's progress regarding a given task, to decide whether that task should be auctioned again or not, the adjustment of the interval would not be an issue, but, in the implemented MRTA system, the tasks (roles), have no defined duration or terminal states. To overcome this issue, the implemented system expresses the allocation problem as an instantaneous iterated allocation problem, like in [23] and [22], and the adjust of the most suitable interval must be made empirically.

4.3 Reinforcement Learning Algorithm - HAQ(λ)

Each robot on the team runs its own HAQ(λ) algorithm (seen in Sec. 3), which is used to formulate the bidding towards each of the roles being offered by the auctioneer. The robot's experience is not shared with teammates. To create the notion of team work, though, the reinforcements received by all agents are equal, and given only when a goal is scored or suffered.

The design of the sate space for the RL algorithm is key to effective learning. In the role selection abstraction level, the RL algorithm's state space needs to represent aspects of the dynamics of the environment, such as robot speeds and positioning over time, statistical distributions about passing skills, as well as ball positions over time. This concept of capturing attributes with broader temporal significance appears on other works that operate in similar levels of abstraction, like [20] and [16].

The state space implemented has 27 dimensions. It uses features obtained by an algorithm that keeps a histogram of the last 10 x, y positions of the robots and ball on the field, sampled once per second. At each iteration cycle, this algorithm extracts dynamical characteristics of the robots using the histogram data, such as distance traveled and region of the field where the robot stayed the most, recently. The state space also has features measuring the number of recent kicks to the goal from both teams, and the amount of time the ball stayed in the offensive field.

Even using higher level features, the resulting state is still too large and time to convergence would be prohibitive. Also, the robot cannot be expected to experience all possible states, is has to learn with limited experience and having visited only a sparse sample of the state space. Therefore, the Q-value tables must be approximated using some representation with fewer variables,

a technique known as *function approximation*. In this work, CMACs with tile coding and hashing, implemented similarly to the proposed in [21], are used for function approximation. The CMAC and tile coding detailed description can be found in details in [15].

For each of the available roles, a set of metrics was created to serve as heuristics for the RL algorithm, using programmer's domain knowledge. Mostly, these metrics were extracted from the hand-coded role selection system that existed previously in the team's strategy software, such as logic to evaluate the opponent's positions in the field and determine if there's need for defending the goal. The advantage of the heuristic functions proposed by [1] is that any function capable of producing a scalar output can be used. Also, if the heuristic is incorrect, as the RL algorithm operates and gains experience this heuristic will be superseded.

Reinforcements are given to all robots on the team when a terminal state, defined as a goal in favor or against, is reached. For goals in favor the value of the scalar reinforcement is +100, and -100 for goals suffered. A small negative reinforcement, -1, is given for every transition that does not reach a terminal state (goal). This small reinforcement serves to prevent the robots from learning to do nothing. The heuristic functions are normalized to one order of magnitude less than the reinforcements, between -10 and +10.

The parameters employed on the HAQ(λ) algorithm are: ϵ-*greedy* algorithm, with $\epsilon = 0.1$, $\gamma = 0.9$, $\lambda = 0.3$, and eligibility trace by substitution.

The next section details the experiments performed with the MRTA system and results obtained.

5 Experiments and Results

To validate the proposed MRTA system, four experiments were performed in simulation. Three partial implementations of the system, to be serve as benchmarks, and the actual MRTA system. One short experiment using real robots was also performed, during the –suppressed competition name– 2010.

The experiments, both simulated and with real robots, used two instances of the –suppressed name– 2010 software, one using the MRTA system and the other without it. This software is the same used by the –suppressed name– team during the RoboCup 2010 in Singapore.

The experiments used a computer with two Intel Xeon Quad Core 2.26GHz processors and 32GB RAM memory. The simulated results were performed using the simulator's time acceleration of 30 times, meaning that 1 second in real time corresponded to 30 seconds in simulation, thus allowing each trial to be performed in around 5 days.

5.1 Simulated Experiments

Four simulated experiments, described ahead, were executed. All of them were executed in 5 independent trials, each containing 20,000 games.

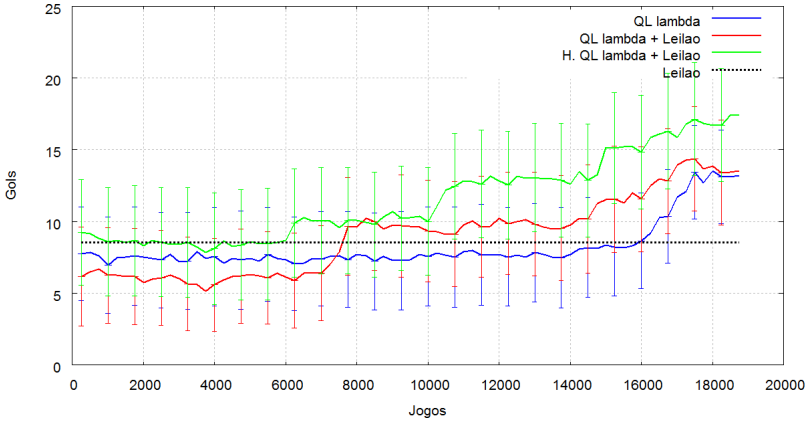

Fig. 2. Comparative of the different MRTA experiment results. The graph shows the average and standard deviation of the goals scored per match, in groups of 10 games, for 5 independent trials. Each trial had 20,000 games, except the Auction-only experiment.

Auction-only Experiment - On this experiment, the auctioning module was used and the robots participated without using reinforcement learning. The robot's utility functions towards each role were the result of the metrics developed to be heuristics. The goal was to find the average goals per match at different intervals between auctions. A notice: as this experiment involved no learning algorithm, each of its trial had only 500 games.

Table 1 shows the results, with the average goals and respective standard deviation. The results of the table indicate that with 5 seconds, the performance is low, mostly because the actions the robots perform are often cut before conclusion by a too short time between auctions. 15 seconds has the best result among the tested intervals, while, for 30 and 45 seconds, a degradation in performance occurs, due to slow response to changes in the game conditions.

Q(λ) Experiment - This experiment consisted in using the Q(λ) algorithm instead of the auctioning module on the Coach agent, thus allowing it to select the roles of all the robots centrally. The Q(λ) was used with the same parameters, $\epsilon = 0.1$, $\gamma = 0.9$ e $\lambda = 0.3$. The interval between interactions of the algorithm was set to 15 seconds or when a terminal state was reached, as in all other experiments. This interval was used for it was the best result of the experiment with different intervals described earlier.

Table 1. Influence of the different interval between auctions (in seconds) on the average goals per match. Table shows average and standard deviation for each case.

Interval (s)	Average goals per match
5	2.06 ± 1.43
15	8.62 ± 3.47
30	6.31 ± 2.92
45	3.35 ± 1.76

This experiment shows how the original Q(λ) algorithm, operating centrally, performs. The result shows that learning occurs, but the time taken for outperforming the hand-coded algorithm is high, as expected.

Auction-Q(λ) Experiment - On this experiment the MRTA system is employed with its two modules, the auctioning on the Coach and the RL module on the robots. Only the heuristics were absent from the algorithm, thus leaving all the computation of utility functions for the Q(λ) algorithm. The initial performance is low, but after approximately 7,000 games the performance is already above the performance of the hand-coded and RL-only algorithms.

Auction-HAQ(λ) Experiment - This experiment uses the proposed MRTA system in full, with the auctioning module being performed by the Coach and the robots executing the Heuristically Accelerated Q(λ) (HAQ(λ)) algorithm. Equation (6) was used to define the value of the function $H(p, s)$, used on (5). This function is simple: the heuristics of all roles p' were calculated for the state s and, for the heuristic with higher value, the result $H(p)$ was used. For all the other roles, the $H(p)$ value was zero.

$$H(p,s) = \begin{cases} H(p,s) \text{ in case } arg\max_{p'}(metric(p',s)) \\ 0 \quad \text{otherwise} \end{cases} \tag{6}$$

The result of all the experiments is shown in Fig. 2, from where some observations can be made. The first of them is that the use of the Q(λ) only leads to a considerably longer learning curve, what was expected. It can be observed also that the Auction-Q(λ) experiment has an initial average of 6.2 goals per match, inferior to both the Q(λ) only and Auction only experiments, which are 8.62 and 7.50, respectively. This happened because the RL algorithm has no initial domain knowledge, so it bids random values on the auctions, what results in deficient allocations. However, after some time, the performance becomes superior to the experiment without RL. On the case of the Auctions-Q(λ), after 7,000 games the result is above the Auction only, and after 17,000 games, the average was already above 13 goals, an increase of more than 50% in comparison Auction only experiment.

5.2 Experiments with Real Robots

The experiment with real robots is an attempt to explore transfer what was learned during simulation, as it is not feasible to execute the number of games needed by the RL algorithm to converge using real robots. The experiment thus consisted of copying the the Q-values contained in the CMACs and other learning parameters of the HAQ(λ) algorithm, after the execution of different amounts of simulated games. These memory blocks were then used in matches in laboratory between real robots. Table 2 shows the results of 3 independent trials of 50 games each, averaged. These results have trends similar to the simulated ones, what indicates that, in fact, learning from simulated environment can be used on the real robots.

Table 2. Result of the copying of RL knowledge gained during simulation to real robots, compared to the performance of the Base team, without any MRTA system.

Algorithm	Sim. games transferred	Average goals (real robots)
Base team (without MRTA)	-	3.12 ± 1.52
Auction-HAQ(λ)	0	3.01 ± 2.34
	5000	5.20 ± 3.11
	10000	7.69 ± 3.65
	15000	7.44 ± 3.04

One empirical test was also made during an official RoboCup game in 2010, when the team executed in the real robots the MRTA system with the knowledge gained during 20,000 simulated games.

Fig. 3 shows extracted logs of this game, where the –suppressed name– team is the yellow. In Fig. 3(a), the delineated areas in white represent where ball and adversaries have been in the last few seconds, according to the histograms described in Sec. 4.3.

A full description of the logs shown in Fig. 3 is the following: (a) the moment an auction occurs and one of the defense robots takes an Attacker role, while its teammate heads to the goal, in possession of the ball. (b) A kick to the goal occurs. (c) Blue team's goalkeeper defends, bouncing the ball back. Meanwhile, the yellow robot who just changed roles heads to the ball from the defensive field. (d) Ball rolls towards the middle of the field, and the mentioned yellow robot approaches the ball. The robot takes the ball and kicks again to the goal.

Although not resulting in a scored goal, the logs show an example situation where the MRTA system gave extra offensive strength to the team. The authors acknowledge this is an evidence rather than a proof. Nevertheless, it is an evidence in line with the results of the experiments performed in simulation and laboratory.

The next section brings the conclusions of this paper.

6 Conclusions and Future Work

The results of the experiments indicate that the proposed MRTA system was capable of improving the performance of the team, and that the union of market-based methods and reinforcement learning resulted in superior performance than their separated usage, what can be seen on the Fig. 2 graph.

The results also show that the heuristic acceleration could overcome the initial stage deficiency of RL algorithm, as the heuristics provide the initial domain knowledge the RL lacks. The heuristically accelerated algorithm also demonstrated the best results, another evidence in favor of the use of heuristics in the RL algorithm.

Nevertheless, even with the heuristic acceleration, the reinforcement learning algorithm still has convergence times too high for use directly. However, it is very useful for training the task allocation system off-line, outperforming the

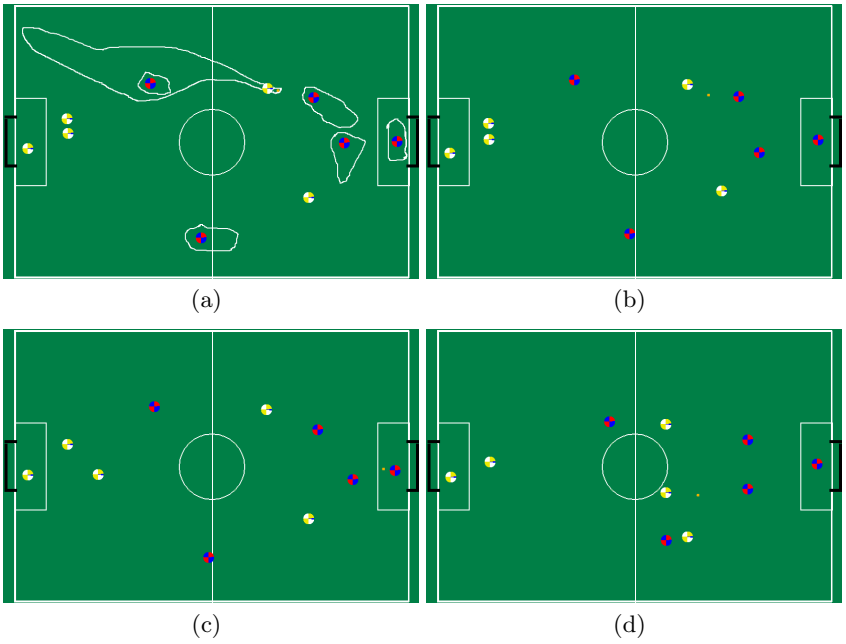

Fig. 3. Logs of the match (between real robots) on the 2010 competition. The –suppressed– team is shown in yellow and white.

hand-coded approaches. This prior training can become particularly powerful if a solution for modeling the opponent's behavior is used. There are works on this regard, although the authors have not researched into it as of the writing of this paper.

The authors also believe that more research into the topic of transfer learning for RL domains [19] could considerably improve the capabilities to apply what was learned in simulation to real robots.

References

1. Bianchi, R.A.C., Ribeiro, C., Costa, A.: Accelerating autonomous learning by using heuristic selection of actions. Journal of Heuristics 14, 135–168 (2008)
2. Browning, B., Bruce, J., Bowling, M., Veloso, M.: STP: Skills, tactics and plays for multi-robot control. IEEE Journal of Control and Systems Engineering 219, 33–52 (2005)
3. Bruce, J., Zickler, S., Licitra, M., Veloso, M.: Cmdragons: Dynamic passing and strategy on a champion robot soccer team. In: Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA). Pasadena, CA (2008)
4. Celiberto, L., Ribeiro, C., Costa, A., Bianchi, R.A.C.: Heuristic reinforcement learning applied to robocup simulation agents. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007: Robot Soccer World Cup XI. pp. 220–227. Springer Berlin / Heidelberg (2008)

5. Dias, M.B., Zlot, R.M., Zinck, M.B., Gonzalez, J.P., Stentz, A.T.: A versatile implementation of the traderbots approach for multirobot coordination. In: Int. Conf. on Intelligent Autonomous Systems (2004)
6. Dias, M., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7), 1257–1270 (July 2006)
7. Gerkey, B., Matarić, M.: Sold!: auction methods for multirobot coordination. Robotics and Automation, IEEE Transactions on 18(5), 758–768 (2002)
8. Gerkey, B., Matarić, M.: Multi-robot task allocation: analyzing the complexity and optimality of key architectures. Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE Int. Conf. on 3, 3862–3868 vol.3 (Sept 2003)
9. Gerkey, B.P., Matarić, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. Int. Journal of Robotics Research 23(9), 939–954 (2004)
10. Kose, H., Tatlidede, U., Mericli, C., Kaplan, K., Akin, H.L.: Q-learning based market-driven multi-agent collaboration in robot soccer. In: Proceedings of the Turkish Symposium on Artificial Intelligence and Neural Networks. pp. 219–2228 (2004)
11. Kyrylov, V.: Balancing gains, risks, costs, and real-time constraints in the ball passing algorithm for the robotic soccer. In: Lakemeyer, G., Sklar, E., Sorenti, D., Takahashi, T. (eds.) RoboCup-2006: Robot Soccer World Cup X. pp. 304–313. Springer Verlag (2007)
12. Parker, L.E., Tang, F.: Building multirobot coalitions through automated task solution synthesis. Proceedings of the IEEE 94(7), 1289–1305 (July 2006)
13. Parker, L.E.: Distributed intelligence: Overview of the field and its application in multi-robot systems. Journal of Physical Agents 2(1), 5–14 (March 2008), special issue on Multi-Robot Systems
14. Sandholm, T., Suri, S.: Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In: Proceedings of the Seventeenth National Conf. on Artificial Intelligence. pp. 90–97 (2000)
15. Stone, P., Sutton, R.S., Kuhlmann, G.: Reinforcement learning for RoboCup-soccer keepaway. Adaptive Behavior 13(3), 165–188 (2005)
16. Sukthankar, G., Sycara, K.: Robust recognition of physical team behaviors using spatio-temporal models. In: AAMAS '06: Proceedings of the fifth Int. joint Conf. on Autonomous agents and multiagent systems. pp. 638–645. ACM (2006)
17. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge,MA. (1998)
18. Tang, F., Parker, L.E.: A complete methodology for generating multi-robot task solutions using asymtre-d and market-based task allocation. Robotics and Automation, 2007 IEEE Int. Conf. on pp. 3351–3358 (April 2007)
19. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research 10(1), 1633–1685 (2009)
20. Vail, D., Veloso, M.: Feature selection for activity recognition in multi-robot domains. In: AAAI'08, Twenty-third Conf. on Artificial Intelligence (2008)
21. Watkins, C.J.C.H.: Learning from Delayed Rewards. Ph.D. thesis, University of Cambridge (1989)
22. Weigel, T., Auerbach, W., Dietl, M., Dumler, B., Gutmann, J.S., Marko, K., Muller, K., BernhardNebel, Thiee, B.S.M.: CS Freiburg: Doing the right thing in a group. In: Stone, P., Kraetzschmar, G., Balch, T. (eds.) RoboCup 2000. Springer (2001)
23. Werger, B., Mataric, M.J.: Broadcast of local eligibility for multi-target observation. In: 5th Int. Symposium on Distributed Autonomous Robotic Systems (DARS). pp. 347–356 (2000)