Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

Using Multi-Agent Options to Reduce Learning Time in Reinforcement Learning

Danielle M. Clement and Manfred Huber
Computer Science and Engineering
University of Texas at Arlington
Arlington, TX USA
danielle.clement@mavs.uta.edu huber@cse.uta.edu

Abstract

Distributed multi-agent learning has recently received significant interest but also proven to be very complex as the decisions made by any individual agent are not the only factors in the outcomes of those decisions. Uncertainty associated in the decisions and exploration choices of other agents add complexity and delay to individual learning processes. To address this complexity and provide for better scaling of distributed multi-agent learning this paper extends the options framework and the Nash-Q learning technique to apply to multi-agent distributed learning in non-cooperative game theoretic settings. We illustrate the effectiveness of this approach in a grid world and demonstrate improved learning for a set of tasks in a semi-cooperative environment.

Introduction

Control of multi-agent systems and in particular using reinforcement learning in order to learn control strategies for such systems has recently gained significant popularity and interest to facilitate real world applications. However, distributed multi-agent learning of equilibrium strategies can be very complex and time consuming, making it difficult to scale to real-world tasks. Part of the complexity arises because in a multi-agent environment, at any given timestep, an individual agent's decision is not the only contributing factor to the result of its actions, and consideration and prediction of the choices made by other agents slows the learning process and adds uncertainty to individual agent learning. This complexity is increased when exploration is introduced, as the exploration of other agents adds complexity to the learning calculations and updates of individual agents. For this reason, many implementations of game-theoretic multi-agent approaches handle decision making in a centralized fashion. This, however, is generally not realistic since it requires either a centralized coordinator or a completely cooperative scenario where one of the agents can determine the actions of all other agents. In most practical task domains neither of these assumptions is realistic since there is generally no central coordinator and agents are not fully cooperative (e.g. in a robotics domain, even if the overall task were collaborative, each agent would normally also have its own objective to not be damaged which is not shared with the other agents). To successfully address general multi-agent domains it is thus essential to be able to address distributed, non-collaborative decision making and to be able to scale the decision learning to larger, more real world tasks.

In single agent systems, Sutton, Precup and Singh introduced the concept of options, or extended-time-step actions, to reduce the complexity of learning in a single agent system (Sutton, Precup, and Singh 1999). To achieve a similar scaling effect in multi-agent domains, this paper extends this concept to multi-agent systems, where a learning-time benefit is shown in the coordination mechanism established by providing multi-agent options. When agents select multi-agent options to execute as their equilibrium strategies, the equilibrium strategy character of these options implies that those agents have agreed that for the duration of that option, they will follow a set joint policy.

Options have previously been extended to consider a single agent executing multiple policies concurrently (multi-tasking) (Rohanimanesh et al. 2001), to task allocation domains and to transfer learning research. Many applications of options to multi-agent systems focus on applying options traditionally, as single agent policies in a multi-agent environment, such as in Riedmiller and Merke's soccer 'moves' (Riedmiller and Merke 2002), or on leveraging a centralized control agent with decision making authority. Examples of applications of options to multi-agent systems include the work by Stone, Sutton, and Kuhlmann (Stone, Sutton, and Kuhlmann 2005) on a keepaway subtask of robot soccer, Ghavamzadeh and Mahadevan's (Ghavamzadeh and Mahadevan 2004) application of options to communication and coordination hierarchies, and

Trigo and Coelho's (Trigo and Coelho 2007) joint-intention multi-agent option model.

Stone, Sutton, and Kuhlmann (Stone et al. 2005) focus on keepaway (a subtask of robot soccer). While those agents use options to coordinate passing behavior among teammates, they are single-agent options and the selection of those options is carried out by only one teammate (only the ball-possessor selects an option). Using collaborative options to support joint decision making is not addressed.

Ghavamzadeh and Mahadevan (Ghavamzadeh and Mahadevan 2004) extend the options framework to multi-agent systems in a purely cooperative task. Their work focuses on using an options-informed framework to determine when agents would be best served to coordinate and retrieve information about other agents' intended action choices in a hierarchical task structure, but does not focus on team behaviors other than to maximize total system utility. Their work is extended by Cheng, Shen, Liu, and Gu (Cheng et al. 2007) to learn options at multiple levels in the task hierarchy, but that work also does not look at options to facilitate teamwork, but rather as a mechanism to maximize the total system utility. Recent work in Dec-POMDPs (Amato, Konidaris, and Kaelbling 2014) also extends the options framework to multi-agent systems, and similarly focuses on purely cooperative domains.

Trigo and Coelho (Trigo and Coelho 2007) describe options as a set of intentions for individual systems, and describe a hierarchical approach where the intentions of individual agents are coordinated into the intentions of the total group of agents. At each decision epoch in their model, the agent can choose whether to select an action individually or request that one be selected for them by the collective. The collective and individual policies are motivated by different goals and are learned in parallel. The primary difference between their work and the structure we are about to describe is the selection of opportunities to collaborate via a centralized collective versus the selection of specific collaboration behaviors by individual agents.

Options have proven to be an efficient mechanism to speed learning time for single agents by Provost, Kuipers, and Miikkulainen (Provost, Kuipers, and Miikkulainen 2004) who compared learning effectiveness in a robot-sensing environment between options-enabled and primitive action learners. They concluded that options-enabled learners were able to rapidly identify critical areas of the state space because options moved high-value, distant rewards through the learning system more efficiently.

In this paper, we briefly cover the techniques which have been extended by this research, describe the approach taken to develop joint multi-agent options in general, non-cooperative settings, describe an experiment on learning speed and action selection in a grid world with collaborative and competitive aspects, and present conclusions.

Background

In the work described in this paper we will extend two primary concepts, options and Nash-Q.

Options

Options, as described by Sutton, Precup and Singh (Sutton et al. 1999), provide a mechanism in single agent systems to extend actions over multi-step time scales. Formally, options (π, τ, *I*) are described as being comprised of:

- a policy π: $S \times A_p \rightarrow [0,1]$ where S is the set of all states and A_p is the set of primitive actions, where π represents the probability of the corresponding action;
- a termination condition τ: $S \times A_p \rightarrow [0,1]$ where S is the set of all states and A_p is the set of primitive actions, where τ represents the probability of terminating the option;
- an initiation set $I \subseteq S$.

When in an initiation state, an agent has the ability to select an option associated with that initiation state. Once an option has been selected, an agent will follow the policy associated with that option until that option terminates per τ. Primitive, or single-step, actions can also be defined as options, and combined with the SMDP work of Bradtke and Duff (Bradtke and Duff 1995), the Sutton, Precup and Singh paper defines policies over options, thereby enabling temporal hierarchies of behavior. Using this hierarchical SMDP learning has been shown to have the potential to significantly accelerate learning and to allow for larger problems to be learned than could be done using only primitive actions. (Provost et al. 2004)

Nash-Q

Nash-Q is a reinforcement learning algorithm that extends Q-learning to the game-theoretic multi-agent domain. Since in these domains greedy utility maximization by each agent is no longer a viable solution, Nash-Q learns instead a game-theoretic equilibrium strategy. In particular, Nash-Q learning for multi-agent systems (Hu and Wellman 2003), uses the Nash Equilibrium as the update feature for Q-learning as opposed to the max. This was chosen as a mechanism to address the fact that learning in multi-agent systems is challenging due to the uncertainty associated with other agents' explorations and that an agent who possesses some knowledge about the choices, benefits or rewards of other agents can improve their learning by making decisions informed by that knowledge in addition to reasoning about their own behavior.

A Nash-Q learner learns in a distributed fashion and reasons that the other agents within a system will behave rationally (in a game-theoretic sense) in future interactions, and therefore selecting an equilibrium behavior will prove more advantageous in semi-cooperative games than selecting an independent max behavior.

Nash-Q, formally, extends the traditional Q-value update equations for reinforcement learning:

$$Q_{t+1}(s_t, a_t) = (1 - \alpha_t) * Q_t(s_t, a_t) + \alpha_t \left[R_t + \beta \max_{a} Q_t(s_{t+1}, \mathrm{a}) \right] \tag{1}$$

where α_t is a learning rate and β represents a discount factor and R_t is the learner's reward at time t to stochastic multi-agent games as

$$Q_{t+1}^{i}\big(s_t, a^1, \dots, a^n\big) = (1 - \alpha_t) * Q_t^{i}\big(s_t, a^1, \dots, a^n\big) + \alpha_t\big[R_t^{i} + \beta\, NashQ_t^{i}(s')\big] \quad (2)$$

where $NashQ_t^i(s')$ is the expected payoff of the Nash equilibrium strategy of the stage game associated with the next state Q-values. Selection of that equilibrium strategy consistently across agents is one of the challenges of Nash-Q. For the initial implementation, Hu and Wellman selected the first equilibrium.

Nash-Q learning is a commonly used learning algorithm in multi-agent domains and we extend it to the situation where agents can use multi-agent options as action choices.

Approach

In this paper we will expand on the concept of an option from single agent reinforcement learning to allow for hierarchical learning with extended time actions in multi-agent domains. For this we will extend the concept of the single agent option to the game theoretic multi-agent setting and integrate it into a hierarchical multi-agent learning framework. We will illustrate the effect this has on the value function update in the Nash-Q learning algorithm and will finally show the benefit of this in a grid world domain.

A multi-agent option extends a single-agent option as follows. A multi-agent option (P, π, τ, I) is comprised of:

- a set of participating players $P \subseteq X$, where X is the set of all known agents
- a multi-agent policy (i.e. an equilibrium strategy profile) in the form of a set of policies (one for each participating player) $\pi_{1\dots n}\colon S \times A_{1\dots n} \rightarrow [0,1]$ where S is the set of all states, A_p is the set of primitive actions, and n is the number of participating players;
- a termination condition $\tau\colon S^{+} \times A_{1\dots n} \rightarrow [0,1]$ where S^{+} is the set of all states including any termination states, A_p is the set of primitive actions, and n is the number of participating players;
- an initiation set $I \subseteq S$.

The primary extension from the standard option framework is the shift from a single agent policy to a set of policies (one for each participating player) which jointly form a Nash equilibrium strategy for a given task represented by the multi-agent option. When in an initiation state, each player has the ability to select a multi-agent option. If all participating players select the multi-agent option, for as long as all participating players jointly pursue that option, those agents follow a previously established collaborative interaction based on execution of their own individual policies π_i. None of the agents should have an incentive to terminate outside the joint termination condition because once agents have decided to pursue the "subtask" represented by the option (and do not change their mind), the fact that the associated policies form a Nash equilibrium strategy should remove any incentive to deviate from this strategy. Following this reasoning, for the focus of this paper, agents terminate execution of a multi-agent option either (a) when that option reaches a probabilistic termination state or (b) when it is detected that the other agents are not participating in the selected option. This detection happens immediately after the first action is taken within an option, as the current implementation is fully observable. Other termination schemes for agent non-selection or early exit from an option choice will be addressed in future work.

At each decision epoch, the players use Nash-Q to determine the Nash equilibrium action they will play for the next cycle. This selected action could be a primitive action or a multi-agent option. If both players choose to play the same multi-agent option, they are effectively choosing to collaborate for the duration of that option. For the purposes of this work, Nash equilibrium are identified by the GAMBIT game theory tool suite (McKelvey, McLennan, and Turocy 2011).

An individual player P_i's decision process is as follows:

1. P_i selects an action (either primitive or multi-agent option) for the current state s per the learning method used (in this case Nash-Q, but the process can be generalized). In this case, the selection process is per Nash-Q, and will satisfy a Nash equilibrium strategy when the available actions to consist of all primitive options and all multi-agent options where s is a member of that option's initiation set and P_i is a member of the set of participating players for that option.
2. All players execute their selected actions and receive observations. If a player has selected a multi-agent option, the primitive action to execute is selected based on the player's policy π_i associated with that multi-agent option.
3. P_i updates the Q-value of the current state s for the primitive actions that were taken.
4. If all participating players are executing a multi-agent option, and that option has terminated, P_i updates the Q-value for the initiation state of that option with the discounted cumulative reward associated with executing that option.
5. If P_i is executing an option and all participating players are also executing that option, P_i's next action for the new state is selected from the associated policy π_i and return to step 2. Otherwise (including the case where all participating players are not executing the option), return to step 1.

Primitive actions are updated per the standard Nash-Q process, but updating the value of the options is handled slightly differently. After an option o has completed, the value of its initiation state s_0 is updated per Eq. 3.

$$Q_{t+1}(s_0, o) = [(1 - \alpha) * Q_t(s_0, o)] + \left[\textstyle\sum_{\mathrm{i}=0}^{u} \beta^{i} R_i\right] + \big(\alpha * \beta^{u+1} * NashQ(s')\big) \quad (3)$$

where u represents the number of timesteps the option was active, and R_i is the reward received in intermediate state s_i, which is a state traversed during option execution.

For the purposes of this paper, the assessment of whether or not all participating players have selected an option is handled through the full-observability of the system – all players know via observations exactly which (if any) multi-agent option was selected by each independent agent and which primitive action was executed immediately after the first action is executed. Future work will include the assessment of whether or not all participating players are executing a multi-agent option as well as the termination responses if agents leave a multi-agent option prematurely.

Design of Experiments

To demonstrate the operation and assess the benefits of the presented hierarchical learning approach using multi-agent options with Nash-Q, a set of experiments in a two-agent grid-world domain that contains collaborative and competitive task elements was performed.

The target world for these experiments is the 3x3 grid world illustrated in Fig. 1. The purpose of the game is for each player to locate one of two keys and bring it to their goal to open a player-specific door to a blocked region. Each key can only be picked up by a single agent when both agents occupy the same square as the key. To access the keys, the agents must cross a bridge square, which they can only enter together. If an agent attempts to enter the bridge square alone, that agent remains in place. If both agents attempt to pick up a key, neither agent gets the key. The game ends when at least one door is opened with the key, meaning that the agent and a key are in the goal square and a specific action is taken to open the door. Each agent is penalized for every time step, making the final score dependent on the time it takes to retrieve and use a key. This penalization as well as the difference in reward between the agent that picks up a key and the agent that only assists comprises the competitive aspect of the game. Agents receive a small bonus reward for holding a key and a slightly smaller reward for the other agent holding a key. There is a large reward associated with opening the door. All rewards aggregate at each time step.

Table 1. Definition of the initiation states and termination probabilities associated with the six multi-agent options

Option	Initiation State Definition	Termination Probability Definition
Move Up To Bridge	$P_1y = 0$ & $P_2y = 0$ & $\sim(P_1x = 1$ & $P_2x = 1)$	1 if $P_1x = 1$; $P_2x = 1$; $P_1y = 1$; $P_2y = 1$ 0 otherwise
Move Down To Bridge	$P_1y = 0$ & $P_2y = 0$ & $\sim(P_1x = 1$ & $P_2x = 1)$	1 if $P_1x = 1$; $P_2x = 1$; $P_1y = 1$; $P_2y = 1$ 0 otherwise
Player 1 Gets Key 1	$P_1y = 2$ & $P_2y = 2$ & $K_1p =$ NONE & $K_2p \neq P_1$	1 if $K_1p = P_1$ 0 otherwise
Player 1 Gets Key 2	$P_1y = 2$ & $P_2y = 2$ & $K_1p \neq P_1$ & $K_2p =$ NONE	1 if $K_1p = P_1$ 0 otherwise
Player 2 Gets Key 1	$P_1y = 2$ & $P_2y = 2$ & $K_1p =$ NONE & $K_2p \neq P_2$	1 if $K_1p = P_2$ 0 otherwise
Player 2 Gets Key 2	$P_1y = 2$ & $P_2y = 2$ & $K_1p \neq P_2$ & $K_2p =$ NONE	1 if $K_1p = P_2$ 0 otherwise

In this environment, the states are described by the positions of each player, the positions of each of the keys, and for each key, which (if any) agent possesses the key. These parameters describe the available state space S: P1x × P1y × P2x × P2y × K1x ×K1y ×K2x ×K2y ×K1p ×K2p; where P1 and P2 represent the players in the game, K1 and K2 represent the keys, x and y represent x and y positions in the grid world environment [0,2], and p represents the possession of a given key {NONE, P1, P2}. This results in 59,049 distinct states.

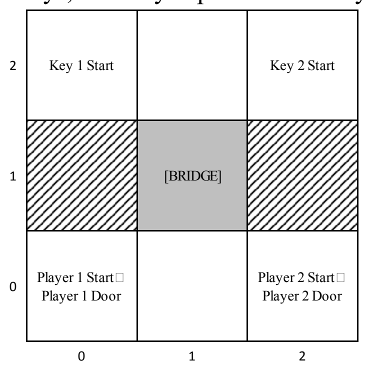

Figure 1. The Grid World

There are six available primitive actions Ap {UP, DOWN, LEFT, RIGHT, PICKUP, and DROP}. Overall, there are also six available multi-agent options O {Move Up To Bridge, Move Down To Bridge, Player 1 Gets Key 1, Player 1 Gets Key 2, Player 2 Gets Key 1, and Player 2 Gets Key 2}. In each of these multi-agent options the joint policy represents a Nash equilibrium strategy for obtaining the associated task objective indicated in the name of the option. Both players must participate in these options (P = {P1, P2}). Rules defining the initiation states and the termination probabilities associated with these options are defined in Table 1. All states that meet the criteria indicated in Table 1 are members of the initiation set for that multi-agent option.

A selected example path for two agents from the "move up" option is illustrated in Fig. 2. Each of the players has the ability to select a multi-agent option in the associated start states for that option in addition to the primitive ac-

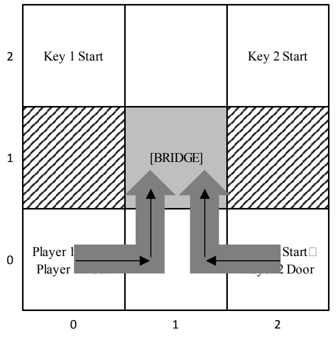

Figure 2. A sample path taken by two agents executing the Move Up To Bridge option

tions. No options are available when agents are split (one in the top row and one in the bottom row) or in the bridge square itself. The policies associated with the options represent Nash equilibrium strategies for their specific associated subtasks and, while they have been hand-coded in these experiments, could easily be learned policies via a sub-goal learning process.

The options themselves are each focused on an aspect of the game that requires coordination among the players to be successful, for example, both players being in the same square as a key with only one player attempting to pick the key up. This results in some sub-optimal individual behaviors for the agents when seen in relation to their final task rewards. For example, if an agent is already in the same square as a key, the other agent is not, and a “get key” option is jointly selected, the agent that is already positioned correctly will remain in that location (at individual cost) until the other agent arrives at that position.

If a player selects a multi-agent option and the other player does not select the same option, the player will terminate option execution and reselect based on Nash-Q. This is enabled by the fully observable nature of the current implementation and allows our experiment to remain unconcerned with detecting if the other player has selected an option. If both players select the same multi-agent option, that option executes until a common termination happens per the termination probabilities.

To determine learning performance, these experiments were run for 10,000 episodes. Games were run with or without multi-agent options available. Execution of an episode continued until a player opened a door or 500 primitive actions were executed. In the learning period, players would randomly select an option per an exponentially decaying rate (λ) of -0.0005, which would transition to a 1% exploration rate (ϵ) after the learning period was completed (episode 8400). Initial Q-values are set to 5, which is the reward when both players hold keys, without the movement penalty. In the case where there were multiple equilibria, agents would select risk dominant equilibrium (Harsanyi and Selten 1988). If there were multiple risk dominant equilibrium, the agents would select the risk dominant equilibrium that maximized their multiplicatively combined utility.

Results

The score of both agents over the course of their learning is shown in Fig. 3, with the multi-agent-options-enabled agents represented with dark lines and the options-free agents with gray lines. Indications of the time during the experiments when the exploration rate reaches 0.1 and 0.01 are shown by the black diamonds. The agents with multi-agent options available are able to learn the optimal policy of both agents opening their doors simultaneously. The agents without multi-agent options available under the same learning time parameterization are only able to learn the sub-optimal policy of both agents retrieving keys under the same parameters. One of the key differences in our work to previous implementations of options in multi-agent systems is that each agent makes decisions based on individual utility. As seen in these results, the agents with multi-agent options learn the optimal policy together. Despite the fact that at least one of the non-options-enabled agents is able to reach the goal during the exploration period, those agents stabilize to a sub-optimal policy when the exploration period is concluded, because at least one agent does not have any learned incentive to cooperate beyond retrieving the keys.

As illustrated in Fig. 4, multi-agent options make more of the policy space (represented as a percentage of the q-table explored) accessible to agents quickly. Agents are able to use multi-agent options to leap forward to less easily accessible regions of the solution space and exploit the rewards available there. This result replicates the single agent domain findings of Provost, Kuipers, and Miikkulainen (Provost et al. 2004) in a multi-agent domain.

As shown in Fig. 5, agents independently selected to

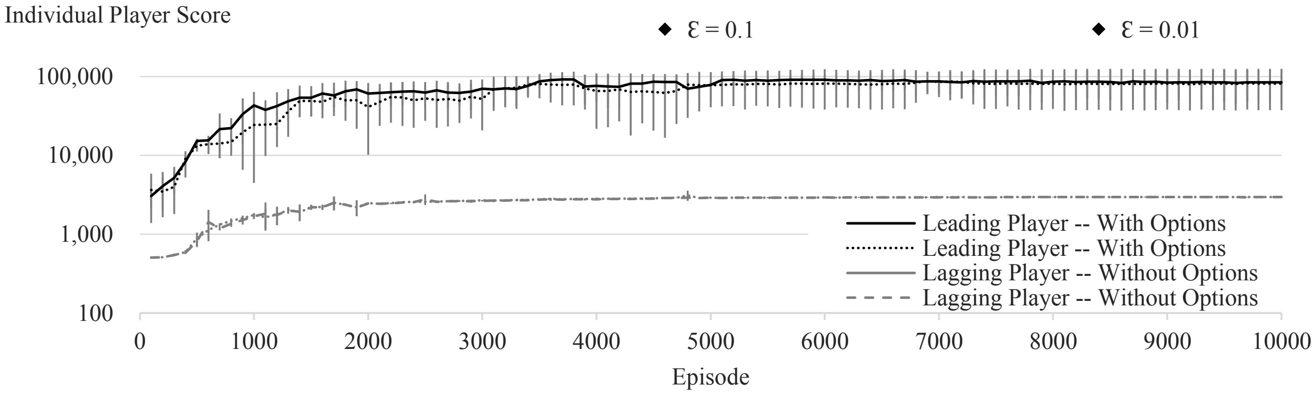

*Figure 3. Comparison of the score for each agent in a two agent game with and without multi-agent options. Each point represents the average score for 100 episodes across 5 runs. The bars represent +/- one standard deviation. The exploration rate decreases exponentially from 1 at a rate of -0.0005*episode, to a minimum of 0.01. The black diamonds illustrate where the exploration rate (ϵ) = 0.1 and 0.01. Note these values are plotted logarithmically, and all values have been increased by a constant (1000) to ensure that they are depicted on this scale. After each run, the leading player was established as the player with the highest average score over the last 100 episodes.*

collaborate via options a greater percentage of time as they learned. The spike shown near episode 8000 is due to a single run with some late exploration on the part of the lagging player. The variance in option selection is due to the fact that in some runs, after an initial learning period, the primitive actions within the option policies replaced the options themselves. Since options and their underlying actions initially resulted in the same q-values, and since the update process was asymmetric (Q-values of primitive actions were updated as options executed, but options were not updated when primitive actions aligned with option-based policies executed), primitive actions have a slight learning advantage. In a few cases, once a policy was learned, the behavior was to "dissolve the option" and use the constituent actions instead. However, it is important to note that while the final policy no longer uses the options, it performs the same actions. The agent's early use of options during learning is essential for the overall learning process, including the learning of the Q-values of the primitive actions used in the final policy.

Conclusions

In this paper, we have shown that multi-agent options in a hierarchical reinforcement learning framework establish an early advantage to learning collaborative techniques. The pre-established coordination mechanisms available through these options establish a structure for agents to reach areas of the solution space that would be limited by the exploration of other agents.

Extending this research to support learned collaborative options rather than generated options is a logical extension of this work. Also, further analysis could be done to compare the performance of multi-agent collaborative options to traditional single agent options. This analysis could determine whether the contribution of multi-agent options was due to the collaboration inherent in those options, or to the extended timescale associated with those options. The collaborative nature of these subtasks would make development of these options difficult without some maintenance of history, since single agent options could get caught in loops if the other agents were uncooperative (for example, by trying to simultaneously retrieve a key, thereby blocking both agents).

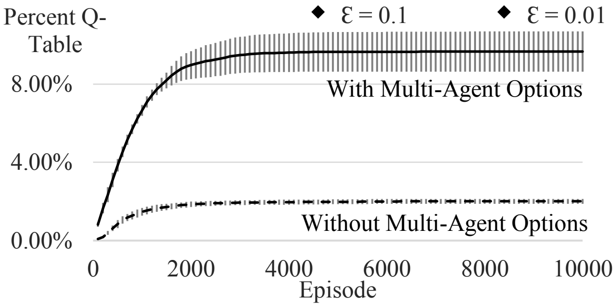

Figure 4. Percentage of Q-Table explored with multi-agent options and without multi-agent options.

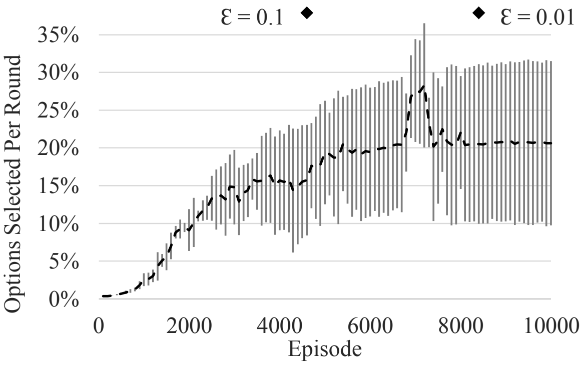

Figure 5. Options selected per executed round.

Exploring other selection criteria for equilibrium strategies or implementing other learning algorithms (other than Nash-Q) within this framework would also be of interest as an extension to this work.

References

Amato, C., Konidaris, G., & Kaelbling, L. 2014. Planning with macro-actions in decentralized POMDPs. *AAMAS 2014* (pp. 1273–1280). Int. Foundation for Autonomous Agents and Multiagent Systems.

Bradtke, S. J., & Duff, M. O. 1995. Reinforcement learning methods for continuous-time Markov decision problems. *Advances in Neural Information Processing Systems*, 7(7): 393.

Cheng, X., Shen, J., Liu, H., & Gu, G. 2007. Multi-robot cooperation based on hierarchical reinforcement learning. *ICCS 2007* (pp. 90–97). Springer.

Ghavamzadeh, M., & Mahadevan, S. 2004. Learning to communicate and act using hierarchical reinforcement learning. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3* (pp. 1114–1121). IEEE Computer Society.

Harsanyi, J. C., & Selten, R. 1988. *A general theory of equilibrium selection in games.* (1st ed.). MIT Press Books.

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

McKelvey, R. D., McLennan, A. M., & Turocy, T. L. 2011. Gambit: Software Tools for Game Theory, Version 0.2010.09.01.

Provost, J., Kuipers, B., & Miikkulainen, R. 2004. Self-organizing perceptual and temporal abstraction for robot reinforcement learning. *AAAI-04 wkshp learning and planning in markov processes* (pp. 79–84).

Riedmiller, M., & Merke, A. 2002. Using machine learning techniques in complex multi-agent domains. *Adaptivity and Learning*.

Rohanimanesh, K., Mahadevan, S., Rohanlmanesh, K., & Lansing, E. 2001. Decision-theoretic planning with concurrent temporally extended actions. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence* (pp. 472–479). Morgan Kaufmann Publishers Inc.

Stone, P., Sutton, R. S., & Kuhlmann, G. 2005. Reinforcement Learning for RoboCup-Soccer Keepaway. *Adaptive Behavior*, 13(3): 165–188.

Sutton, R., Precup, D., & Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1): 181–211.

Trigo, P., & Coelho, H. 2007. A hybrid approach to teamwork. *Proc. VI Encontro Nacional de Inteligncia Artificial (ENIA-07).*