The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Socially Augmented Hierarchical Reinforcement Learning for Reducing Complexity in Cooperative Multi-agent Systems

Xueqing Sun, Laura E. Ray, Jerald D. Kralik, Dongqing Shi

Abstract— **This paper addresses the inherent complexity in coordinating learned behavioral strategies of multiple agents working towards a common goal. Because of the interactions among the agents, a primary challenge of policy learning is escalating computational complexity with increasing number of agents and the size of the task space (including action choices and world states). We employ an approach that incorporates social constructs based on analogs from biological systems of high functioning mammals in order to constrain state-action choices in reinforcement learning. Additionally, we use state-space abstraction and a hierarchical learning structure to improve learning efficiency. Theoretical results bound the reduction in computational complexity due to state abstraction, hierarchical learning, and socially-constrained action selection in learning problems that can be described as decentralized Markov decision processes. Simulation results show that these theoretical bounds hold and that satisficing multi-agent coordination policies emerge, which reduce task completion time, computational cost, and memory resources compared to learning with no social knowledge.**

I. Introduction

There is ample evidence in the natural world that mammals learn to solve complex problems arising in domains of interest to both mammals and teams of robots, namely cooperative hunting/tracking, foraging, and patrolling of territories. These classes of problems can be described mathematically as some variant of a decentralized Markov decision process (Dec-MDP), which are proven to have non-deterministic exponential (NEXP) complexity [1, 2]. Considering the role of social intelligence in sophisticated problem-solving behaviors observed within these domains leads to fundamentally new approaches to deriving intelligent multi-agent systems (MAS) for complex tasks, while managing computational complexity.

In this paper, we focus on decentralized cooperative systems of heterogeneous agents with full observability working towards a common goal. To solve the Dec-MDP, we take the reinforcement learning (RL) approach [3] in which a group of agents interact with their environment through trial-and-error learning to maximize a performance measure in the long run based on reward or punishment received from the environment.

Manuscript received March 10, 2010. This work was supported by the Office of Naval Research under Grant No. N00014-08-1-0693.

Xueqing Sun and Laura E. Ray are with the Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 USA (e-mail: {xueqing.sun, laura.e.ray}@dartmouth.edu).

Jerald D. Kralik and Dongqing Shi are with the Department of Psychological & Brain Sciences, Dartmouth College, Hanover, NH 03755 USA (e-mail: {jerald.d.kralik, dongqing.shi}@dartmouth.edu).

Many attempts have been made to address complexity challenges that derive from a large state/action space, limited bandwidth or lack of communication between agents, uncertainties, and limited perception [4-9]. The hierarchical learning structure of [5] incorporates a task decomposition approach in which the overall value function of a core MDP is decomposed into value functions based on individual subtasks. While task decomposition allows learning to focus on a subtask and ignore other parts of the state space, the system designer must manually identify subgoals and define subtasks that achieve these subgoals. Ghavamzadeh et al. [6] present another hierarchical reinforcement learning (HRL) framework for cooperative multi-agent tasks in which the levels of hierarchy include *cooperative subtasks*, and the coordination among agents in these subtasks significantly improves performance. Again, prior knowledge of the task structure and inter-agent communication is required. In [7-9] social conventions and roles are used to restrict the action choices of agents prior to action selection for large scale problems. A simple social convention can rely on the state or a unique ordering of agents and actions. For example, the traffic law to state who has the right-of–way is a useful convention to prevent conflicts and accidents.

Given challenges arising from complexity, we draw from observations that high functioning mammals solve complex problems cooperatively and efficiently. For example, both Bottlenose dolphins and chimpanzees form shifting and nested alliances, the former during collaborative fishing and the latter in hunting and territorial patrols [10, 11]. Alliances are based on social factors, such as kinship, dominance hierarchy and previous history [10, 11]. In both species, a first-order coalition almost invariably includes two to three individuals that can form a higher-order, larger team of 4 to 14 individuals that varies in stability, size and relatedness. Such team forming makes task differentiation and collective learning more efficient, especially when individuals have complementary skills.

Another theme from biological systems is that often a *satisficing* solution - a mathematical concept denoting an acceptable, albeit suboptimal solution - is adopted based on the rational that a feasible solution to a complex task is better than no solution when cost of obtaining information or of the learning process itself is high [12]. Although many underlying abilities required for complex problem solving in intelligent mammals appear to be innate, it is clear that learning is required and that social intelligence and satisficing play an important role in improving learning efficiency in decision making. For example, Tai chimpanzees take on several roles in a sophisticated strategy when hunting Colobus monkeys [11]. Prey in the canopy are

selected by the hunting party from the ground. The *driver* chimp climbs the occupied tree and begins pursuit, moving the prey forward. Other chimpanzees move along the ground, anticipating the prey's heading. *Blockers* move into the canopy at strategic points, deterring the prey from moving in those directions and funneling the prey towards a trap. The *ambusher* must anticipate the direction and speed of the prey and must determine which tree to climb at what time to intercept the prey. The ambusher either catches the prey, turns the prey back toward other team members, or forces the prey to the lower canopy, where the chimpanzees are faster than the prey. Each role is learned over time, with the most difficult, ambusher role taking up to 20 years to master. Thus, older, experienced male chimpanzees take the ambusher role, and role specialization is consistent with male dominance hierarchy, age, and experience.

Field and laboratory research has also shown that primates are endowed with cognitive abilities for tracking social information, such as recognizing individuals, identifying kin, assessing the qualities of prospective allies, and knowing the nature of third-party relationships, such as rank relationships [13], all of which contribute to efficient hunting, foraging, and patrolling strategies.

Drawing on themes from [10-13], this paper makes two key contributions to decentralized multi-agent learning that diverge from traditional task allocation algorithms. First, we leverage key social constructs in RL to solve decentralized MAS problems efficiently. Second, we use an abstract state space representation and hierarchical learning structure to reduce complexity. Social information makes policy searching more tractable, and abstract state representation reduces the state-action space, thereby reducing the number of state-action pairs that are considered, as well as memory resources. Taken together, these components provide satisficing solutions to otherwise intractable MAS learning problems.

The paper is organized as follows. Section II summarizes Dec-MDP and RL complexity. Section III describes socially augmented Q-learning and derives theoretical results placing bounds on the reduction in complexity achievable through state abstraction, hierarchical learning, and social knowledge. Section IV presents empirical results demonstrating complexity reduction in a multi-agent foraging task.

II. Dec-MDP Complexity & Reinforcement Learning

A Markov decision process (MDP) is a mathematical framework for solving sequential decision-making problems in stochastic domains. A Dec-MDP of a multi-agent system is formally defined as a 4-tuple $\langle S, \{A_i\}, P, \{R_i\} \rangle$. S is a set of states $s \in S$; $\{A_i\}$ is a finite action set available to agent i with $a_i \in A_i$; $P(s' \mid s, a_i, a_{-i})$ is a table representing the probability of transitioning from state s to state s' due to action a_i taken by agent i and a_{-i} denoting the actions taken by all other agents but i; and R_i is the reward function for agent i. Figure 1 shows a general state-action transition diagram. The black dots represent actions taken by an agent, and squares indicate the next possible states that are also affected by other agents' actions. Dashed nodes will be explained in section III.

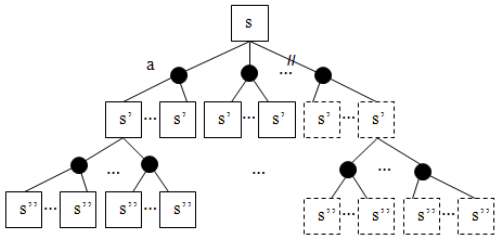

Fig. 1 General state-action transition diagram

Computing the optimal solution for a general Dec-MDP is proven intractably hard in [1, 14]. A complexity analysis of a Dec-MDP is given in this section for multiple heterogeneous agents collaboratively foraging through goal oriented behaviors. Before considering the MAS, we start with single agent foraging where the agent collects one of n scattered entities at each episode k in a finite horizon with transition probability=1. At each episode k, an agent has n-k action choices (tasks to select from), and the size of the state-action table (a measure of complexity) is

$$\mathbb{C}(s,a) = \prod_{k=0}^{n-1}(n-k) = n! \,. \tag{1}$$

The decision version of eq. 1 is similar to the well-known Travelling Salesman Problem [15]. It is NP-complete, and the worst-case run time for an algorithm to solve problems in this class increases exponentially with n. Extending from single agent foraging to a *centralized* MAS with m agents, the complexity remains O($n!$):

$$\begin{aligned}\mathbb{C}(s,a) &= \prod_{k=0}^{n/m-1}(n-km)(n-km-1)...(n-km-m+1) \\ &= \prod_{k=0}^{n/m-1}\frac{(n-km)!}{(n-km-m)!} = n!\end{aligned} \tag{2}$$

The reason behind this result is that, in centralized systems, a single agent (supervisor) makes decisions for all agents, who act as remote slaves, or every agent thinks alike.

Moving up the complexity scale, Bernstein [1] and Goldman [14] prove that the finite-horizon Dec-MDP of m-agents with indirect communication is NEXP-complete. Without loss of generality, we introduce the following notation as a measure of complexity of the Dec-MDP. Let $\varphi_t[i]$ be the number of actions that can be taken by agent i at episode t, $\chi_{tk}[i]$ be the number of possible resulting states that can be encountered by agent i caused by its own k_{th} action and other agents' unknown actions, and $n[i]$ be the number of episodes by agent i, with $\sum_{i=1}^{m} n[i] = n$ for m agents. Using this notation, a complexity metric based on the size of the state-action table in a general Dec-MDP is defined as

$$\mathbb{C}(s,a_i) = \sum_{t=0}^{n[i]-1}\sum_{k=1}^{\varphi_t[i]}\chi_{tk}[i] \,. \tag{3}$$

Complexity in Dec-MDPs increases dramatically due to asynchronous action choices from other agents and the number of possible states. We show in section III how state abstraction, hierarchical learning, and social knowledge together can reduce computational complexity by many orders of magnitude.

Q-learning is a reinforcement learning algorithm based on the updating of the state-action value Q at the end of a sequence of time steps as shown in eq. 4 [16]. The best found policy is then recorded in π^*:

$$Q_t(s,a) = (1-\alpha_t)Q_{t-1}(s,a) + \alpha_t[r + \gamma \max_{a'}\{Q_{t-1}(s',a')\}]$$
$$\pi^* = \arg\max_a Q(s,a) \quad (4)$$

s is the current state at time step t, a is the action taken at state s, r is the reward received from the environment $\alpha \in (0,1]$ is the learning rate, and $\gamma \in [0,1]$ is the discount factor. The convergence of single agent (or centralized) *Q*-learning has been proven in [17, 18] with conditions of action-value pairs visited indefinitely often and α_t satisfying

$$\lim_{T\to\infty}\sum_{t=0}^{T}\alpha_t = \infty \text{ and } \lim_{T\to\infty}\sum_{t=0}^{T}\alpha_t^2 < \infty. \quad (5)$$

In addition to asynchronous learning with simplicity and speed, *Q*-learning is an anytime algorithm, which means the algorithm can be interrupted at any time and will return the best policy found thus far. This feature makes *Q*-learning a rational choice in applications where there is a constraint on computation cost. However, it is well known that when single agent (or centralized) *Q*-learning is applied to MAS, convergence properties are not guaranteed [19, 20]. For decentralized MAS, [21] proposes a distributed *Q*-learning algorithm by solving a local RL problem that updates an optimistic agent's policy by neglecting punishment caused by non-cooperative behaviors from other agents. They prove that this algorithm will converge to the unique optimal solution in a deterministic cooperative MAS. However, when multiple optimal joint actions exist, neither convergence nor bounds are guaranteed.

III. Socially Augmented Hierarchical Q-learning

Our reinforcement learning approach to solving Dec-MDPs uses social constructs, state abstraction, and hierarchical learning to reduce complexity and to achieve a satisficing solution to problems whose solution is otherwise intractable.

A. Incorporating Social Hierarchy Constructs

Based on social intelligence in biological domains reviewed in section I, we propose a social structure representation illustrated in Fig. 2 that codes social hierarchy for use in multi-agent RL. It consists of three levels from the bottom up: *Physical level*, *Association level* and *Task Force level*. The *Physical level* can have several parallel physical feature hierarchies based on mobility, sensor resolution and other individual capabilities, with robots ranked in each category. The *Association level* includes two major components - a role hierarchy and relational matrices. Within the role hierarchy, agents are associated and ranked in various role types, based on information from the physical level, and relational matrices define social relationships, such as affiliation and kinship. Thus, for example, agents ranked higher on mobility and sensor resolution may be ranked higher for a specific role based on a weighted sum; and stable, first-order coalitions of a few individuals may be coded within relational matrices. The *Task Force level* emerges based on information from the lower *Association level* for specific tasks. For instance, in foraging, the agents are scored and drafted based on their ranking in various roles along with their relationship to other agents. Compared to static organization, the *Task Force level* is where dynamics and flexibility are involved in multi-agent team structuring for different missions.

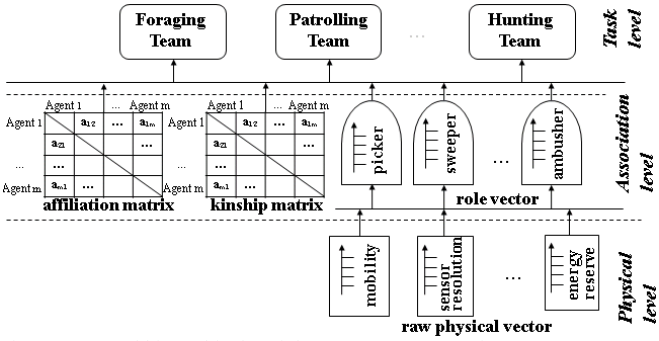

Fig. 2. Proposed hierarchical social structure representation

The potential benefits of using a social structure to represent social knowledge in a multi-agent cooperative task include performance-based team configuration, conflict resolution, and complexity reduction in teaming and tasking through hierarchical and social relationships [22].

B. Abstract Representation of State Space

To improve tractability, we build an abstract state representation so each agent can learn its role in a hierarchical learning framework in advance of task selection and task execution [23, 24]. Taking a foraging task as an example, a team of heterogeneous robots learn to collaborate in sweeping, collecting and depositing scattered entities to a home location to reduce task completion time**.** Robots of type P2 can pick up two objects or less at once; and robot type S_P1 can pick up a single object only, but can also emerge as a sweeper S when needed to cluster objects, i.e., move one object next to another object so a P2 robot can pick them up simultaneously. The abstract state for each agent i includes the number of each type of entity, e.g., number of single objects g and clusters c; the number of each type of robot n(P2) and n(S_P1), and a boolean variable μ_i indicating whether there is a higher ranking robot in the team of agent i:

$$\mathbb{S} = \{g, c, \text{n(P2)}, \text{n(S\_P1)}, \mu_i\} \quad (6)$$

Abstracting from individual entities and robots to the count of entities and robots is a natural reflection of how the mammalian brain codes and tracks information at the lowest representational level necessary to complete a task.

Although abstraction may reduce the likelihood of finding a traditional optimal solution, e.g., minimum time or distance traveled, the size of the state space is significantly reduced compared with a state representation that retains additional characteristics, such as location of objects in the environment. Moreover, learned policies generalize more readily to similar problems. For example, if a P2 robot exists in the team, a S_P1 robot is likely to emerge as a sweeper whether there are 10, 20, 50, or 100 single entities on the field, as it is more effective for the sweeper to sweep objects into clusters for the P2 robot to collect than to pick up single objects. Agents select their roles dynamically during learning based on the Boltzmann distribution of the role values (section III.C).

A generous upper bound on the size of the state space for the abstract state defined by eq. 6 and m agents and n entities is $(m+1)\times(m+1)\times(n-d+1)\times(d+1)\times 2$ where d is the number of clusters at $t = 0$. To derive a bound in terms of m and n, let $N = n + 1$, then $(n-d+1)\times(d+1) = (N-d)(d+1)$. Since $N^2 - 4Nd + 4d^2 = (N-2d)^2 \geq 0$, it follows that $(N-d)d \leq N^2/4$, and $(N-d)(d+1) \leq N^2/4 + N - d$. Since $\forall d$, $(N-d)(d+1) < N^2/4 + N$, an upper bound for the size of the abstract state space is $2(m+1)^2\left[\frac{(n+1)^2}{4} + n + 1\right]$. For the *centralized* multi-agent system without state abstraction, eq. 2 shows $O(n!)$ complexity; with $m = 5$ and $n = 50$ as an example, a reduction from 3e64 to 50490 through state abstraction is achieved.

C. *Socially Augmented Hierarchical Q-learning*

Learning in a large state space can be done cost-effectively through both in-depth and lateral development of a hierarchy, which, in general, is an ordered set or an acyclic graph consisting of nodes and branches of their subordinates. During state space exploration, by selecting one branch in the hierarchy we reduce the computational complexity by ignoring other branches and their sub-branches. Figure 3 illustrates this hierarchical learning structure for foraging.

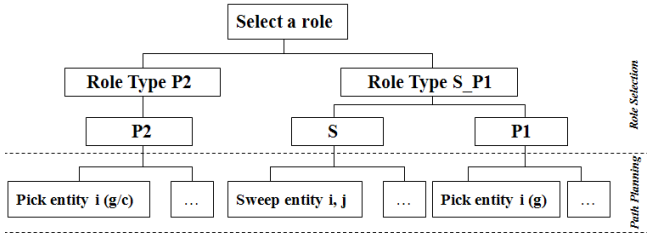

Fig. 3. Hierarchical learning structure for foraging example

As seen in Fig. 3, the first layer of learning includes a role selection node branching out to various related behaviors, followed by a learning layer for path planning. Within path planning, the action space is constrained to the selected role, which reduces the number of action choices and space for exploration downstream. For example, once a robot selects a sweeper role, it has a limited action set for learning high level policies of which entities to sweep. After these entities are identified, a lower level in the hierarchy is associated with assumed lower level behaviors, such as navigating to the entities and actuating the sweeping action. This layer can be either a learned or fixed closed-loop policy for navigation.

For decentralized RL, we adapt the idea of Distributed Q-learning [21], which places less importance on penalties received due to other agents' bad action choices, causing exploration to converge to the optimal equilibrium for coordination. The update equation is given by

$$Q_i(s,a_i) = \begin{cases} Q_i(s,a_i) + \alpha\Delta & \text{if } \Delta \geq 0 \\ Q_i(s,a_i) & \text{otherwise} \end{cases} \quad (7)$$

where $\Delta = r + \gamma \max_{a_i'}\{Q_i(s',a_i')\} - Q_i(s,a_i)$. This algorithm is guaranteed to converge when there is a single optimal equilibrium; however, if multiple equilibria exist, guaranteed convergence is lost. To deal with this issue, we add a temporal term η_{a_i} in the action selection equation to select the most recent behaviors yielding maximum Q value:

$$\pi_i^* = \arg\max_{a_i}\{Q(s,a_i) + w\eta_{a_i}\} \quad (8)$$

η_{a_i} is the trial index of the action taken, and w is a weight, set to some small value to control $w\eta_{a_i}$ so it will not overtake $Q_i(s,a_i)$. η_{a_i} is a temporal indicator in the trial-and-error learning to store the index of the most recent trial in which an action is taken. It is equivalent to imposing a social convention for agents to choose the latest action set with the maximum Q value, yielding a coordinated action choice in the presence of multiple optimal solutions. For example, a vignette of two robots picking objects to clear the field is shown in Fig. 4. Assuming they are ranked the same and the two objects are the same distance from each robot, there are two sets of optimal joint actions: robot 1 picks object 1 and robot 2 picks object 2; or robot 1 picks object 2 and robot 2 picks object 1. Assuming a decentralized system with no direct communication between robots, to each robot, picking either object 1 or object 2 yields the same equilibrium Q value after many trials based on eq. 7. Without η_{a_i} in eq. 8, the "best" action each robot chooses can be mismatched by chance (e.g., both robots pick object 1), and therefore the optimal joint action is not guaranteed. With η_{a_i} in eq. 8, a social convention is imposed for each robot to choose the best action taken tagged with the largest trial index, which prevents mismatch and yields the best coordinated joint actions.

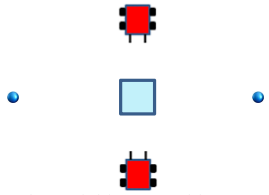

Fig. 4. Vignette of two robots picking two objects

The learning algorithm is summarized in TABLE I. Social knowledge makes an individual agent's role selection dependent on their standing in the role vector of the social structure. For example, if a robot is ranked higher in the sweeper role than the picker role, it is more likely that a sweeper role will be chosen based on a selected distribution, e.g., Boltzmann distribution. Second, social knowledge reduces complexity and resolves conflict based on perception by filtering out tasks that a higher ranking robot is working on or is expected to work on. The second vignette in the accompanying video illustrates this case: a lower ranking robot detects a higher ranking robot in its proximity approaching the same object, ends its current action promptly, and chooses the next action, improving the overall performance by saving otherwise wasted time. From the previous complexity analysis, if we assume the number of superior agents of agent i is $h_t[i]$ at episode t, then the *reduced* state-action space size can be calculated as

$$\overline{\mathbb{C}}(s,a_i)=\sum_{t=0}^{n[i]-1}\sum_{k=1}^{\varphi_t[i]-h_t[i]}\chi'_{tk}[i] \quad (9)$$

where $\{\chi'_{tk}[i]\}$ is a subset of $\{\chi_{tk}[i]\}$, a result of $h_t[i]$ subtrees being pruned from the state space as illustrated by the removed branches (dashed nodes) in Fig. 1. To calculate theoretical bounds on reduction, we have for agent i,

$$\delta[i]=\left(\sum_{t=0}^{n[i]-1}\sum_{k=1}^{\varphi_t[i]}\chi_{tk}[i]-\sum_{t=0}^{n[i]-1}\sum_{k=1}^{\varphi_t[i]-h_t[i]}\chi'_{tk}[i]\right)\le\sum_{t=0}^{n[i]-1}h_t[i]\mathrm{X}_t[i] \quad (10)$$

where $\mathrm{X}_t[i]=\max_k\{\chi_{tk}[i]\}$. The physical interpretation of $\mathrm{X}_t[i]$ is the maximum number of possible resulting states under each action of agent i at episode t. Across m agents, the maximum total state-action space reduction is

$$\sum_{i=1}^{m}\delta[i]\le\sum_{i=1}^{m}\left(\sum_{t=0}^{n[i]-1}h_t[i]\mathrm{X}_t[i]\right)\le\sum_{i=1}^{m}\left(H_{\max}n[i]\mathrm{X}_{\max}\right)=nH_{\max}\mathrm{X}_{\max} \quad (11)$$

where $H_{\max}=\max_{t,i}\{h_t[i]\}$ and $\mathrm{X}_{\max}=\max_{t,i}\{\mathrm{X}_t[i]\}$. Similarly a lower bound is provided by

$$\sum_{i=1}^{m}\delta[i]\ge nH_{\min}\mathrm{X}_{\min} \quad (12)$$

where $H_{\min}=\min_{t,i}\{h_t[i]\}$, $\mathrm{X}_{\min}=\min_{t,i}\{\mathrm{X}_t[i]\}=\min_{t,i}\{\min_k\{\chi_{tk}[i]\}\}$

Finally, social knowledge encourages gregarious behavior during path planning by forming loose leader-follower relationships based on role vector, affiliation and kinship matrices. It has been shown in sociobiology that gregarious mammals have elaborate social links that are useful in foraging, defending themselves, and raising their young [25]. In MAS, gregarious behavior can emerge over time from learning based on higher rewards received from greater compatibility and a reduced state space when a loosely formed sub-team works together in a sub-area of the environment.

IV. Simulation Experiments

To evaluate its effectiveness, we apply socially augmented, hierarchical reinforcement learning with state abstraction to robot foraging. We compare the performance and computational complexity of our approach against baseline experiments in which traditional hierarchical Q-learning, also with state abstraction, is performed without social knowledge, while keeping all other parameters (field configuration, learning rate and number of Monte Carlo experiments) the same. The goal of both decentralized learning algorithms is to search for the action sequence of each robot that yields the lowest overall team completion time in removing all objects from the field. To further demonstrate the learned policy, we use a physically realistic Webots simulator [26] shown in Fig. 5 and in the accompanying video. We do not include centralized Q-learning without state abstraction in the empirical results because the centralized solution cannot be found in reasonable time for the number of entities and agents in the simulation. We assume agents obtain information about entities and other agents through observation, an indirect form of communication. No other communication, including two-way communication, is assumed.

TABLE I
Socially Augmented Q-learning Algorithm

Algorithm: $Q_i^{(r)}(s,a_i)$ and $Q_i^{(p)}(s,a_i)$ is the state-action Q value of role selection and path planning for agent i, r is the reward received, $\alpha\in(0,1]$ is the learning rate, $r\in[0,1]$ is the discount factor and τ is the temperature in Boltzmann distribution.

Initialize:
- Initialize agent i state at time step t=0.
- For all $s\in S$, $a_i\in A_i$, let role $Q_i^{(r)}(s,a_i)=0$, path $Q_i^{(p)}(s,a_i)=0$
- Initialize learning rate $\alpha=0.99$ and discount factor γ=0.9
- Define social structure by agent i's rank and relationship.

While current object state s_t != completion state s_T

Role Selection:
- Choose role a_i at state s based on the probability by Boltzmann distribution,

$$\Pr\{a_i\}=e^{Q_i^{(r)}(s,a_i)/\tau}/\sum_{b_i}e^{Q_i^{(r)}(s,b_i)/\tau}$$

- Receive reward r dependent on the state and social role vector in the association level. Update $Q_i^{(r)}(s,a_i)$:

$$Q_i^{(r)}(s,a_i)=(1-\alpha)Q_i^{(r)}(s,a_i)+\alpha[r+\gamma\max_{a'}\{Q_i^{(r)}(s',a_i')\}]$$

- Record current best known role policy,

$$\pi_i^{(r)*}=\arg\max_{a_i}\{Q_i^{(r)}(s,a_i)\}$$

Path Planning:
- Filter out objects that a higher ranking robots are working on.
- Based on role policy $\pi^{(r)*}$, choose path a_i (object) at state s
- Receive reward r based on relative object location and leader-follower relationship.
- Update $Q_i^{(p)}(s,a_i)$: $\Delta=r+\gamma\max_{a_i'}\{Q_i^{(p)}(s',a_i')\}-Q_i^{(p)}(s,a_i)$

$$Q_i^{(p)}(s,a_i)=\begin{cases}Q_i^{(p)}(s,a_i)+\alpha\Delta & \text{if } \Delta\ge 0\\ Q_i^{(p)}(s,a_i) & \text{else}\end{cases}$$

- Record current best known path policy,

$$\pi_i^{(p)*}=\arg\max_{a_i}\{Q_i^{(p)}(s,a_i)+w\eta_{a_i}\}$$

End

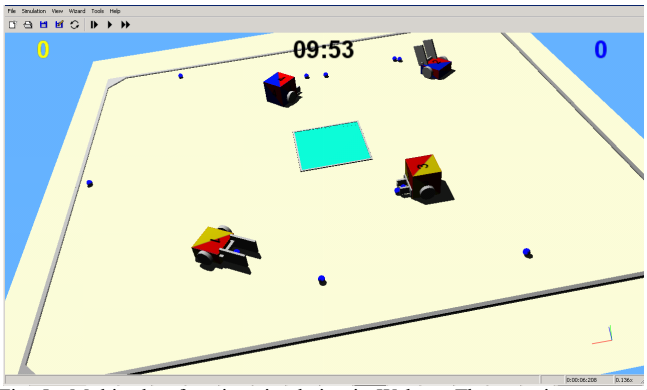

Fig. 5. Multi-robot foraging simulation in Webots: The team is composed of two types of robots: Type I (Robot 1 and 2) can pick up 1-2 objects at a time; Type 2 (Robot 3 and 4) can either sweep or pick up 1 object. Social affiliation is indicated by team colors (Blue/Red or Yellow/Red).

For each team condition (number of entities and number of agents), 200 Monte Carlo experiments are conducted where *n* entities are randomly placed. For every experiment of randomly placed entities, each learning *episode* starts with an agent choosing an action and ends after a reward (non-positive value) is received based on the time it takes to finish the action. A trial ends when all agents run out of action choices, i.e., when all objects are removed from the field. Team performance for each experiment is measured after 10,000 learning trials, although only the first 5,000 trials are shown in accompanying graphs.

We evaluate a series of scenarios involving (1) two heterogeneous robots (one S_P1 and one P2) collecting 20 random entities, (2) five heterogeneous robots (two S_P1s and three P2s) collecting 20 random entities, and (3) five robots (two S_P1s and three P2s) collecting 50 random entities. As shown in the hierarchical learning structure in Fig. 3, at the role selection level, S_P1 robots can select either the picker role or sweeper role. At the path planning level, based on the role selected, robots choose actions such as which entity to pick (e.g., closest to home, closest to myself, or closest to my group leader) or which two single entities to sweep together (e.g., closest two to home, closest two to myself). These actions aim to produce generalized policies for completing the task, rather than policies specific to the initial condition of randomly-placed entities. Further lower level behaviors are assumed, such as a *pick* action that includes a temporally-abstracted action of navigating to the entity, gripping, navigating to home, and releasing.

Figure 6 compares the average team completion time over 200 experiments for both socially augmented Q-learning and ordinary Q-learning. The results show that teams using social knowledge during learning consistently perform better than teams without social knowledge. Note that socially augmented Q-learning, in general, does not explore as many action choices as traditional Q-learning because of preferences regarding roles, conflict avoidance in choosing tasks, or path planning based on social knowledge. Figure 7 shows convergence of the minimum value function over all agents as an overall team completion time trendline, comprised of a five period moving average as a function of trial number, with the ~90% convergence point marked by a red diamond. As seen in Fig. 7, socially augmented Q-learning consistently converges faster than the traditional algorithm due to reduced exploration.

Socially augmented Q-learning is also evaluated and compared to ordinary Q-learning based on the number of distinct states visited. Figure 8 shows the mean number of distinct states visited averaged over 200 Monte Carlo experiments as a function of trial number for each of the five robot scenarios. As seen in Fig. 8, the number of the distinct states visited in socially augmented Q-learning is well below non-social Q-learning, despite the fact that its state space representation has an extra state variable $\mu_t[i]$ to indicate whether there is a higher ranking agent at episode *t*.

As seen in eq. 9, social knowledge reduces the number of action choices during path planning in the amount of $h_t[i]$ at episode *t* alone and causes a cumulative reduction in state-action space complexity. For the 50-entity and 5-robot simulation with ascending ranks (one through five), the maximal number of higher ranking agents is $H_{\max} = 4$ and $H_{\min} = 1$. Also, the maximal number of possible resulting states under any agent *i*'s actions is $X_{\max} = 5 \times 5 = 25$ and $X_{\min} = 1$, i.e., this is how many possible abstract state pairs $\{g, c\}$ four other agents can generate assuming each one takes one or no action in changing the number of single entities *g* and the number of clusters *c*. Therefore, eq. 11 and 12 give an upper bound $50 \times 4 \times 25 = 5000$ and lower bound $50 \times 1 \times 1 = 50$, respectively, on the state-action space reduction. Since it is often the case that each agent's rank descends by one, a tighter upper bound may be obtained using the average $h[i]$ over the team. In this case, the upper bound would be, $50 \times 2 \times 25 = 2500$, which is consistent with the reduction shown in Fig. 8. Note that this reduction is on top of the reduction due to state abstraction.

V. Conclusion and Future Work

In this paper we have explored socially augmented, hierarchical Q-learning with state abstraction to reduce complexity and improve the learning efficiency in solving a Dec-MDP. Social knowledge is embedded in the learning process to improve space searching efficiency. State abstraction and hierarchical learning are employed to make the problem domain more tractable.

Avenues of future work include extending the learning framework to a decentralized partially observable Markov decision process (Dec-POMDP) in which agents do not have full access to the world state due to incomplete or noisy sensory information.

References

[1] Bernstein, D., Givan, R., Immerman, N. & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research, 27 (4), 819-840.

[2] Rabinovich, Z., Goldman, C. V. & Rosenschein, J. S. (2003). The complexity of multiagent systems: The price of silence. Proc. Second Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, 1102-1103, Melbourne, Australia.

[3] Sutton, R. S. & Barto, A. G. (1998). Reinforcement Learning. MIT Press, Cambridge, MA.

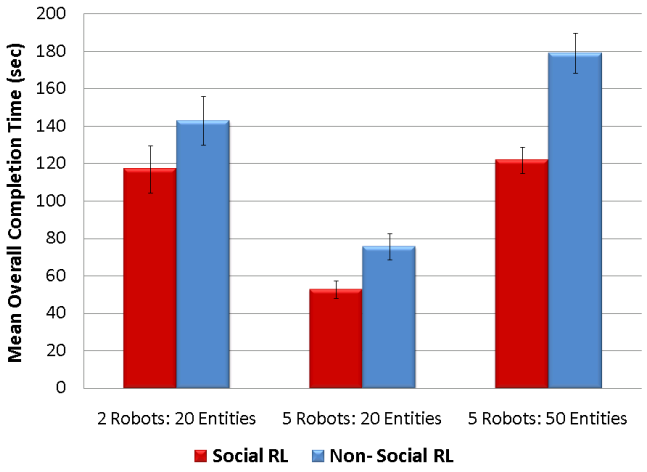

Fig. 6. Overall team completion time in three different scenarios: (1) a heterogeneous team of two robots, one S/P1 and one P2 collecting 20 entities; (2) a heterogeneous team of five robots, two S/P1 and three P2 collecting 50 entities; (3) extension of the same five robots collecting 50 entities.

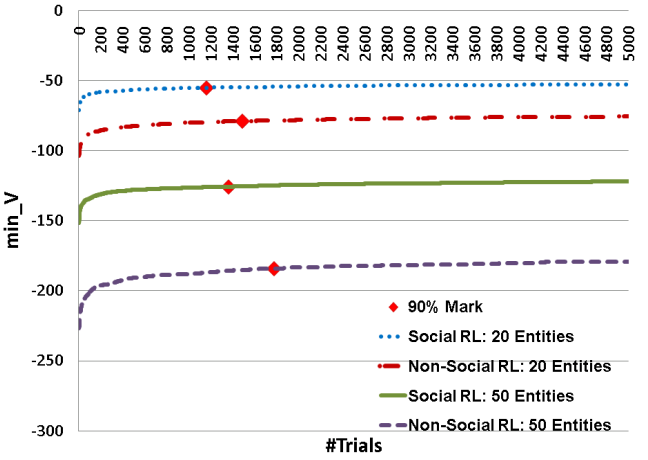

Fig. 7. Comparison of min_V value convergence of traditional (non-social) and socially augmented Q-learning of five robots in two different scenarios: collecting 20 entities and 50 entities.

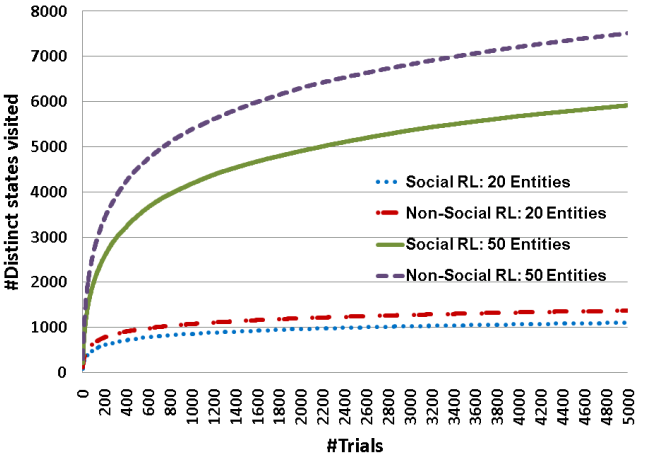

Fig. 8. Number of distinct states visited during traditional (non-social) and socially-augmented Q-learning for five robots in two different scenarios: collecting 20 entities and 50 entities.

[4] Buşoniu, L., Babuška, R. & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Systems, Man and Cybernetics, Part C, vol.38, no.2, pp.156–172.

[5] Dietterich, T. G. (2000). Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition, Journal of Artificial Intelligence Research, vol 13, pp. 227-303.

[6] Ghavamzadeh, M., Mahadevan, S. & Makar, R. (2006). Hierarchical multi-agent reinforcement learning. Autonomous Agents and Multi-Agent System, Volume 13, Issue 2, September 2006, Pages: 197 – 229.

[7] Shoham, Y. & Tennenholtz, M. (1992). On the synthesis of useful social laws for artificial agent societies. In Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, CA, pp. 276-281.

[8] Boutilier, C. (1996). Planning, learning and coordination in multi-agent decision processes. In Proceedings 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-96), De Zeeuwse Stromen, The Netherlands, pp. 195-210.

[9] Spaan, M. T. J., Vlassis, N. & Groen, F. C. A. (2002).High level coordination of agents based on multi-agent Markov decision processes with roles. In Workshop on Cooperative Robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[10] Connor, R. C. (2007). Dolphin social intelligence: complex alliance relationships in bottlenose dolphins and a consideration of selective environments for extreme brain size evolution in mammals. Phil. Trans. R. Soc. B: 362, 587-602.

[11] Boesch, C. (2003). Complex cooperation among tai chimpanzees. In Animal Social Complexity, ed. F. B. M. Waal & P. L. Tyack, pp. 93-110. Cambridge, MA. Harvard University Press.

[12] Byron, M. (1998). Satisficing and Optimality. Ethics 109:67–93. The University of Chicago.

[13] Silk, J. B. (2007). Social Components of Fitness in Primate Groups. Science, vol. 317. no. 5843, pp. 1344 – 1347.

[14] Goldman, C. & Zilberstein, S. (2004). Decentralized control of cooperative systems: Categorization and complexity analysis. Journal of Artificial Intelligence Research 22:143–174

[15] Sanjeev, A. & Boaz, B. (2009). Complexity Theory: A Modern Approach. Cambridge University Press.

[16] Watkins, C. J. C. H. & Dayan, P. (1992). Technical Note: Q-Learning. Machine Learning, Volume 8, Numbers 3-4, pp. 279-292.

[17] Jaakkola, T., Jordan, M. I. & Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. Neural Computation, 6(6), pp. 1185-1201.

[18] Tsitsiklis, J. (1994). Asynchronous stochastic approximation and Q-learning. Machine Learning. 16, pp. 185-202.

[19] Littman, M. L. (1994). Markov games as a framework for multiagent reinforcement learning. In Proceedings of the Eleventh International Conference on Machine Learning, New Brunswick, pp. 157-163.

[20] Claus, C. & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, pp. 746–752.

[21] Lauer, M. & Riedmiller, M. (2000). An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In Proc. of 17th International Conference on Machine Learning (ICML). Morgan Kaufmann, San Francisco, CA. pp. 535–542.

[22] Picault, S. & Collinot, A. (1998). Designing Social Cognition Models for Multi-Agent Systems through Simulating Primate Societies. Proc. of ICMAS'98.

[23] Sun, X., Mao, T., Kralik, J. D. & Ray, L. E. (2009). Cooperative Multi-Robot Reinforcement Learning: A Framework in Hybrid State Space. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE Press, St. Louis, MO.

[24] Mao, T., Sun, X. & Ray, L. E. (2009). Role Selection in Multi-Robot Systems using Abstract State-Based Reinforcement Learning. In the Proceedings of the 14th IASTED International Conference on Robotics and Applications, Boston, MA.

[25] Miller, R. C. (1922). The Significance of the Gregarious Habit. Ecology, Vol. 3, No. 2, pp. 122-126.

[26] Webots Reference Manual. Cyberbotics Ltd. Professional Mobile Robot Simulation Software. http://www.cyberbotics.com.