Multi-agent Reinforcement Learning with Sparse Interactions by Negotiation and Knowledge Transfer

Luowei Zhou*
Robotics Institute
University of Michigan
Ann Arbor, MI, 48105, USA

Pei Yang and Chunlin Chen
Department of Control and Systems Engineering
Nanjing University
Nanjing, Jiangsu, 210093

Abstract

Reinforcement learning has significant applications for multi-agent systems, especially in unknown dynamic environments. However, most multi-agent reinforcement learning (MARL) algorithms suffer from such problems as exponential computation complexity in the joint state-action space, which makes it difficult to scale up to realistic multi-agent problems. In this paper, a novel algorithm named negotiation-based MARL with sparse interactions (NegoSI) is presented. In contrast to traditional sparse-interaction based MARL algorithms, NegoSI adopts the equilibrium concept and makes it possible for agents to select the non-strict Equilibrium Dominating Strategy Profile or Meta equilibrium for their joint actions. For evaluation, we compare the proposed method with several state-of-the-art MARL algorithms in both grid world games and an intelligent warehouse platform, and show that the proposed method can achieve similar performances even without knowing all the joint state-action information.

1 Introduction

Multi-agent learning is drawing increasing interests from scientists and engineers in multi-agent systems (MAS) and machine learning communities [Busoniu *et al.*, 2008]. One key technique for multi-agent learning is multi-agent reinforcement learning (MARL), which is an extension of reinforcement learning in multi-agent domain [Yu *et al.*, 2015]. Several mathematical models have been built as frameworks of MARL, such as Markov games (MG) [Littman, 1994] and decentralized sparse-interaction Markov decision processes (Dec-SIMDP) [Melo and Veloso, 2011]. Markov games are based on the assumption of full observability of all agents in the entire joint state-action space. Several well-known equilibrium-based MARL algorithms [Littman, 1994][Hu and Wellman, 2003][Littman, 2001][Greenwald *et al.*, 2003] are derived from this model. Dec-SIMDP based algorithms rely on agents' local observation, i.e., the individual state and action. Agents in Dec-SIMDP are modeled with single-agent MDP when they are outside of the interaction areas, while the multi-agent model such as MG is used when they are inside. Typical algorithms include LoC [Melo and Veloso, 2009] and CQ-learning [De Hauwere *et al.*, 2010].

In spite of the rapid development of MARL theories and algorithms, more efforts are needed for practical applications of MARL when compared with other MAS techniques [Stone and Veloso, 2000] due to some limitations of the existing MARL methods. The equilibrium-based MARL relies on the tightly coupled learning process which hinders their applications in practice. Calculating the equilibrium (e.g., Nash equilibrium [Nash and others, 1950]) for each time step and all joint states are computationally expensive [Hu *et al.*, 2015b], even for relatively small scale environments with two or three agents. In addition, sharing individual states, individual actions or even value functions all the time with other agents is unrealistic in some distributed domains (e.g., intelligent warehouse [Zhou *et al.*, 2016], sensor networks [An *et al.*, 2011]) given the agents' privacy protections and huge real-time communication costs [Busoniu *et al.*, 2008]. As for MARL with sparse interactions, agents in this setting have no concept of equilibrium policy and they tend to act aggressively towards their goals, which results in a high probability of collisions.

Therefore, in this paper we focus on how the equilibrium mechanism can be used in sparse-interaction based algorithms and a negotiation-based MARL algorithm with sparse interactions (NegoSI) is proposed for unknown dynamic environments. The NegoSI algorithm consists of four parts: the equilibrium-based framework for sparse interactions, the negotiation for the equilibrium set, the minimum variance method for selecting one joint action and the knowledge transfer of local Q-values. Firstly, we start with the proposed algorithm based on the MDP model and the assumption that the agents have already obtained the single-agent optimal policy before learning in multi-agent settings. Then, the agents negotiate for pure strategy profiles as their potential set for the joint action at the "coordination state" and they use the minimum variance method to select the relatively good one. After that, the agents move to the next joint state and receive immediate rewards. Finally, the agents' Q-values are updated by their rewards and equilibrium utilities. When initializing the Q-value for the expanded "coordination state", the agents

*A more detailed version of this paper will be published at IEEE Transactions on Cybernetics [Zhou *et al.*, 2016].

with NegoSI utilize both of the environmental information and the prior-trained coordinated Q-values. To test the effectiveness of the proposed algorithm, several benchmarks are adopted to demonstrate the performances in terms of the convergence, scalability, fairness and coordination ability.

2 BACKGROUND

2.1 Markov Games and MARL

Markov games are widely adopted as a framework for multi-agent reinforcement learning [Littman, 1994] [Hu and Wellman, 2003]. It is regarded as a multiple Markov decision process (MDP) in which the transition probabilities and rewards depend on the joint state-action pairs of all agents. In a certain state, agents' individual action sets generate a repeated game that could be solved in a game-theoretic way. Therefore, Markov game is a richer framework which generalizes both of MDP and repeated game [Nowé *et al.*, 2012].

Definition 1 *(*Markov game*) An n-agent ($n \geq 2$) Markov game is a tuple* $\langle n, \{S_i\}_{i=1,\ldots,n}, \{A_i\}_{i=1,\ldots,n}, \{R_i\}_{i=1,\ldots,n}, T\rangle$*, where n is the number of agents in the system, S_i is the set of the state space for i^{th} agent, $S = \{S_i\}_{i=1,\ldots,n}$ is the set of state spaces for all agents, A_i is the set of the action space for i^{th} agent, $A = \{A_i\}_{i=1,\ldots,n}$ is the set of action spaces for all agents, $R_i : S \times A \to \mathbb{R}$ is the reward function of agent i, $T : S \times A \times S \to [0,1]$ is the transition function.*

Denote the individual policy of agent i by $\pi_i = S \times A_i \to [0,1]$ and the joint policy of all agents by $\pi = (\pi_1, \ldots, \pi_n)$. The Q-value [Sutton and Barto, 1998] of the join state-action pair for agent i under the joint policy π can be formulated by

$$Q_i^{\pi}(\vec{s},\vec{a}) = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_i^{t+k} | \vec{s}^t = \vec{s}, \vec{a}^t = \vec{a}\}, \qquad (1)$$

where $\vec{s} \in S$ stands for a joint state, $\vec{a} \in A$ for a joint action, $\gamma \in [0,1]$ is a parameter called the discount factor and r_i^{t+k} is the reward received at the time step $(t+k)$. Unlike the optimization goal in an MDP, the objective of a Markov game is to find an equilibrium joint policy π rather than an optimal joint policy for all agents. Here, the equilibrium policy concept is usually transferred to finding the equilibrium solution for the one-shot game played in each joint state of a Markov game [Hu *et al.*, 2015b]. Several equilibrium-based MARL algorithms in existing literatures such as NashQ [Hu and Wellman, 2003] and NegoQ [Hu *et al.*, 2015b] have been proposed, so that the joint state-action pair Q-value can be updated according to the equilibrium:

$$Q_i(\vec{s},\vec{a}) \leftarrow (1-\alpha)Q_i(\vec{s},\vec{a}) + \alpha(r_i(\vec{s},\vec{a}) + \gamma\phi_i(\vec{s}'))), \qquad (2)$$

where $\phi_i(\vec{s}')$ denotes the expected value of the equilibrium in the next joint state $\vec{s}'$ for agent i and can be calculated in the one-shot game at that joint state.

2.2 MAS with Sparse Interactions

The definition of Markov game reveals that all agents need to learn their policies in the full joint state-action space and they are coupled with each other all the time [Hu and Wellman, 2003] [Hu *et al.*, 2015a]. However, this assumption does not hold in practice. The truth is that the learning agents in many practical multi-agent systems are loosely coupled with some limited interactions in some particular areas. Meanwhile, the interactions between agents may not always involve all the agents. These facts lead to a new mechanism for MARL, called sparse interaction.

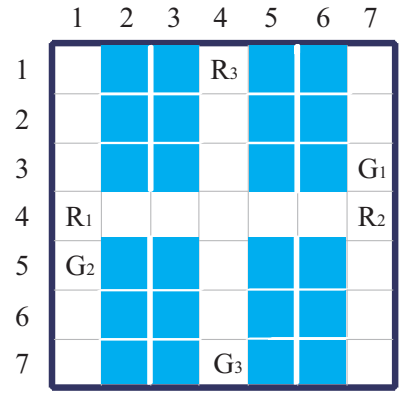

Figure 1: A part of an intelligent warehouse with three robots, where R_i $(i = 1,2,3)$ represents for robot i with its corresponding goal G_i and the shaded grids are storage shelves.

Sparse-interaction based algorithms [Melo and Veloso, 2009][De Hauwere *et al.*, 2010] have recently found wide applications in MAS research. An example of MAS with sparse interactions is the intelligent warehouse systems, where autonomous robots only consider other robots when they are close enough to each other [Nowé *et al.*, 2012] (see Fig. 1), i.e., when they meet around the crossroad. Otherwise, they can act independently.

Earlier works such as the coordinated reinforcement learning [Guestrin *et al.*, 2002] and sparse cooperative Q-learning [Kok and Vlassis, 2004] used coordination graphs (CGs) to learn interdependencies between agents and decomposed the joint value function to local value functions. However, these algorithms cannot learn CGs online and only focus on finding specific states where coordination is necessary instead of learning for coordination [Melo and Veloso, 2009]. Melo and Veloso [Melo and Veloso, 2009] extended the Q-learning to a two-layer algorithm and made it possible for agents to use an additional COORDINATE action to determine the state when the coordination was necessary. Hauwere and Vrancx proposed Coordinating Q-learning (CQ-learning) [De Hauwere *et al.*, 2010] that helped agents learn from statistical information of rewards and Q-values where an agent should take other agents' states into account. However, all these algorithms allow agents to play greedy strategies at a certain joint state rather than equilibrium strategies, which might cause conflicts.

More recently, Hu et al [Hu *et al.*, 2015b] proposed an efficient equilibrium-based MARL method, called Negotiation-based Q-learning, by which agents can learn in a Markov game with unshared value functions and unshared joint state-actions. In later work, they applied this method for sparse interactions by knowledge transfer and game abstraction [Hu *et al.*, 2015a], and demonstrated the effectiveness of the

Algorithm 1 Negotiation-based Framework for Sparse Interactions

Input: The agent i, state space S_i, action space A_i, learning rate α, discount rate γ and exploration factor ε.
Initialize: Global Q-value Q_i with optimal single-agent policy.

for each episode **do**
  Initialize state s_i;
  while true **do**
    Select $a_i \in A_i$ from Q_i with $\varepsilon - Greedy$;
    if (s_i, a_i) is "dangerous" **then**
      Broadcasts (s_i, a_i) and receives (s_{-i}, a_{-i}), form $(\vec{s}, \vec{a})$; /* See Section 3.4 for the definitions of s_{-i} and a_{-i}*/
      if $(\vec{s}, \vec{a})$ is not "coordination pair" **then**
        Mark $(\vec{s}, \vec{a})$ as a "coordination pair" and $\vec{s}$ as a "coordination state", initialize Q_i^J at $\vec{s}$ with Equation 4;
      end if
      Negotiate for the equilibrium joint action with Algorithm 2. Select new $\vec{a}$ with $\varepsilon - Greedy$;
    else {detected an immediate reward change}
      Mark (s_i, a_i) as "dangerous", broadcasts (s_i, a_i) and receives (s_{-i}, a_{-i}), form $(\vec{s}, \vec{a})$;
      Mark $(\vec{s}, \vec{a})$ as "coordination pair" and $\vec{s}$ as "coordination state", initialize Q_i^J at $\vec{s}$ with Equation 4;
      Negotiate for the equilibrium joint action with Algorithm 2. Select new $\vec{a}$ with $\varepsilon - Greedy$;
    end if
    Move to the next state s_i' and receive the reward r_i;
    if $\vec{s}$ exists **then**
      Update $Q_i^J(\vec{s}, \vec{a}) = (1-\alpha) Q_i^J(\vec{s}, \vec{a}) + \alpha(r_i + \gamma \max_{a_i'} Q_i(s_i', a_i'))$;
    end if
    Update $Q_i(s_i, a_i) = (1-\alpha) Q_i(s_i, a_i) + \alpha(r_i + \gamma \max_{a_i'} Q_i(s_i', a_i'))$;
    $s_i \leftarrow s_i'$;
  end while until s_i is a terminal state.
end for

equilibrium-based MARL in solving sparse-interaction problems. Nevertheless, as opposed to single Q-learning based approaches like CQ-learning, Hu's equilibrium-based methods for sparse interactions require a great deal of real-time information about the joint states and joint actions of all the agents, which results in huge amount of communication costs. In this paper, we focus on solving learning problems in complex systems. Tightly coupled equilibrium-based MARL methods discussed above are impractical in these situations while sparse-interaction based algorithms tend to cause many collisions. To this end, we adopt the sparse-interaction based learning framework and each agent selects equilibrium joint actions when they are in coordination areas.

3 NEGOTIATION-BASED MARL WITH SPARSE INTERACTIONS

We decompose the learning process into two distinct subprocesses [Yu *et al.*, 2015]. First, each agent learns an optimal single-agent policy by itself in the static environment and ignores the existences of other agents. Second, each agent learns when to coordinate with others according to their immediate reward changes, and then learns how to coordinate with others in a game-theoretic way. In this section, the negotiation-based framework for MAS with sparse interactions is first introduced and then related techniques and specific algorithms are described in details.

3.1 Negotiation-based Framework for Sparse Interactions

We assume that agents have learnt their optimal policies and reward models when acting individually in the environment. Two situations might occur when agents are working in a multi-agent setting. If the received immediate rewards for state-action pairs are the same as what they learned by reward models, the agents act independently. Otherwise, they need to expand their individual state-action pairs to the joint ones by adding other agents' state-action pairs for better coordination. This negotiation-based framework for sparse interactions is given as shown in Algorithm 1, while the expansion and negotiation process is explained as follows:

1. Broadcast joint state. Agents select an action at a certain state, and they detect a change in the immediate rewards. This state-action pair is marked as "dangerous" and these agents are called "coordinating agents". Then, "coordinating agents" broadcast their state-action information to all other agents and receive corresponding state-action information from others. These state-action pairs with reward changes form a joint state-action and is marked as a "coordination pair". Also, these states form a joint state called a "coordination state", which is included in the state space of each "coordinating agent".
2. Negotiation for equilibrium policy. When agents select a "dangerous" state-action pair, they broadcast their state-action information to each other to determine whether they are staying at a "coordination pair". If so, the agents

need to find an equilibrium policy rather than their inherent greedy policies to avoid collisions. We propose a negotiation mechanism similar to the work in [Hu *et al.*, 2015b] to find this equilibrium policy. Each "coordinating agent" broadcasts the set of strategy profiles that are potential Non-strict Equilibrium-Dominating Strategy Profile (non-strict EDSP) according to its own utilities (See Algorithm 2). If no non-strict EDSP is found, the agents search for a Meta equilibrium set instead, which is always nonempty [Hu *et al.*, 2015b]. Then, if there are several candidates in the equilibrium set, the agents use the minimum variance method to find the relatively good one.

3. When an agent selects an action at a certain state, if neither a change in the immediate reward nor a "dangerous" state-action pair is detected, the agent continue to select its action independently.

3.2 Negotiation for Equilibrium Set

One contribution of this paper is to apply the equilibrium solution concept to traditional Q-learning based MARL algorithms. Different from previous work like CQ-learning and Learning of Coordination [De Hauwere *et al.*, 2010], our approach aims at finding the equilibrium solution for the one-shot game played in each "coordination state". As a result, we focus on two pure strategy profiles [Hu *et al.*, 2015b], i.e., Non-strict Equilibrium-Dominating Strategy Profile (non-strict EDSP) and Meta Equilibrium set. The definition of Non-strict EDSP is as follows.

Definition 2 *(*Non-strict EDSP*): In an n-agent ($n \geq 2$) normal-form game Γ, $\vec{e}_i \in A(i = 1,2,\ldots,m)$ are pure strategy Nash equilibriums. A joint action $\vec{a} \in A$ is a non-strict EDSP if $\forall j \leq n$,*

$$U_j(\vec{a}) \geq \min_i U_j(\vec{e}_i) \tag{3}$$

The negotiation process of finding the set of Non-strict EDSP is shown as in Algorithm 2, which generally consists of two steps: 1) the agents find the set of strategy profiles that are potentially Non-strict EDSP according to their individual utilities; 2) the agents solve the intersection of all the potentially strategy sets and get the Non-strict EDSP set.

However, given the fact that the pure strategy Nash equilibrium (PNE) can be non-existent in some cases, the set of Non-strict EDSP might also be empty. On this occasion, we replace this strategy profile with Meta Equilibrium [Hu *et al.*, 2015b], which is always nonempty.

3.3 Minimum Variance Method

In Section 3.2, we presented the process for all "coordinating agents" to negotiate for an equilibrium joint-action set. However, the obtained set usually contains many strategy profiles and it is difficult for the agents to choose an appropriate one. In this paper a Minimum Variance method is proposed to help the "coordinating agents" choose the joint action with relatively high total utility and minimum utilities variance to guarantee the cooperation and fairness of the learning process. In addition, if the equilibrium set is nonempty, the best solution defined in the minimum variance method always exists.

Algorithm 2 Negotiation for Non-strict EDSPs Set

Input: A normal-form game $\langle n, \{A_i\}_{i=1,\ldots,n}, \{U_i\}_{i=1,\ldots,n}\rangle$.
/* To be noted, "coordinating agent" i only has the knowledge of n, $\{A_i\}_{i=1,\ldots,n}$ and U_i*/
Initialize: The set of non-strict EDSP candidates for "coordinating agents" i: $J^i_{NS} \leftarrow \emptyset$; minimum utility value of PNE candidates for "coordinating agents" i: $MinU^i_{PNE} \leftarrow \infty$; the set of non-strict EDSPs: $J_{NS} \leftarrow \emptyset$.
for each $\vec{a_{-i}} \in A_{-i}$ **do**
  if $\max_{a'_i} U_i(a'_i, \vec{a_{-i}}) < MinU^i_{PNE}$ **then**
    $MinU^i_{PNE} = \max_{a'_i} U_i(a'_i, \vec{a_{-i}})$;
  end if
end for
for each $\vec{a} \in A$ **do**
  if $U_i(\vec{a}) \geq MinU^i_{PNE}$ **then**
    $J^i_{NS} \leftarrow J^i_{NS} \cup \{\vec{a}\}$;
  end if
end for
/* Broadcast J^i_{NS} and corresponding utilities*/
$J_{NS} \leftarrow \bigcap_{i=1}^{n} J^i_{NS}$

3.4 Local Q-value transfer

In most previous literatures [Melo and Veloso, 2009] [De Hauwere *et al.*, 2010], the initial local Q-values of the newly expanded joint states are zeros. Recently, Vrancx et al proposed a transfer learning method [Vrancx *et al.*, 2011] to initialize these Q-values with prior trained Q-value from the source task and got better results. However, Vrancx et al only used coordination knowledge to initialize the local Q-values and overlooked the environmental information, which was proved to be less effective in our experiments. In our approach, we initialize the local Q-values of newly detected "coordination" to hybrid knowledge as follows:

$$Q^J_i((s_i, \vec{s_{-i}}), (a_i, \vec{a_{-i}})) \leftarrow Q_i(s_i, a_i) + Q^{CT}_i(\vec{s}, \vec{a}), \tag{4}$$

where $\vec{s_{-i}}$ is the joint-state set except for the state s_i and $\vec{a_{-i}}$ is the joint-action set except for the action a_i, $Q^J_i((s_i, \vec{s_{-i}}), (a_i, \vec{a_{-i}}))$ is equal with $Q^J_i(\vec{s}, \vec{a})$, $Q^{CT}_i(\vec{s}, \vec{a})$ is the transferred Q-value from a blank source task at joint state $\vec{s}$. In this blank source task, joint action learners (JAL) [Claus and Boutilier, 1998] learn to coordinate with others disregarding the environmental information. They are given a fixed number of learning steps to learn stable Q-value $Q^{CT}_i(\vec{s}, \vec{a})$. Similar to [Vrancx *et al.*, 2011], the joint state $\vec{s}$ in $Q^{CT}_i(\vec{s}, \vec{a})$ is presented as the relative position $(\Delta x, \Delta y)$, horizontally and vertically. When agents attempt to move into the same grid location or their previous locations, these agents receive a penalty of -10. In other cases, the reward is zero.

4 EXPERIMENTS

To test the presented NegoSI algorithm, several groups of simulated experiments are implemented and the results are compared with those of three other state-of-the-art MARL algorithms, namely, CQ-learning [De Hauwere *et al.*, 2010],

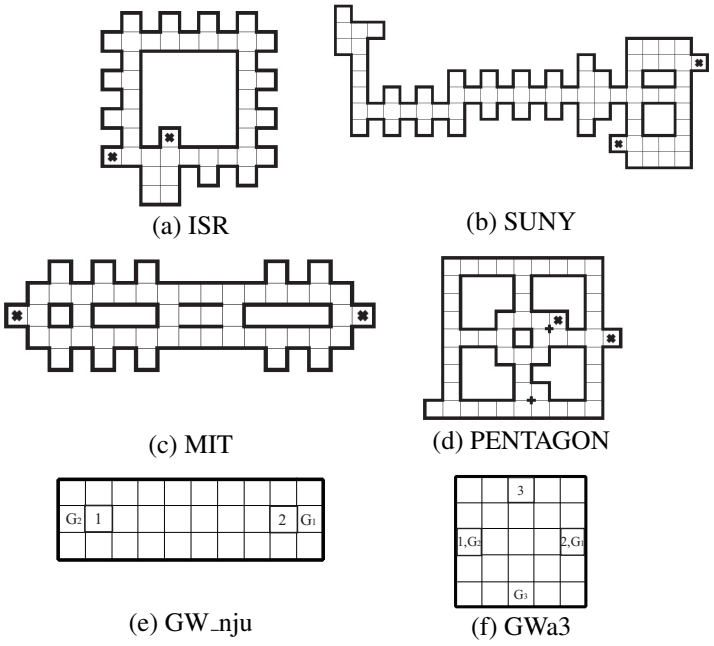

Figure 2: Grid world games.

NegoQ with value function transfer (NegoQ-VFT) [Hu *et al.*, 2015a] and independent learners with value function transfer (IL-VFT).

In our experiments, NegoSI is applied to six grid world games and an intelligent warehouse problem. For all these experiments, each agent has four actions: up, down, left, right. When an agent reaches its goal, it receives a reward of 100. One episode is over when all agents reach their goals. A negative reward of -10 is given when a collision or out-of-boundary occurs. Otherwise, the reward is set to -1 as the power consumption. Hyper-parameters are as follows: learning rate $\alpha = 0.1$, discount rate $\gamma = 0.9$, exploration factor $\varepsilon = 0.01$. All algorithms run 2000 iterations for the grid world games and 8000 for the intelligent warehouse. We evaluate the algorithms with two criteria, i.e., steps of each episode (SEE) and average runtime (AR). All the results are averaged over 50 runs.

4.1 Tests on grid world games

The proposed NegoSI algorithm is evaluated in the grid world games presented by Melo and Veloso [Melo and Veloso, 2009], which are shown in Fig. 2. The first four benchmarks, i.e., ISR, SUNY, MIT and PENTAGON (shown as Fig. 2(a)-(d), respectively), are two-agent games, where the cross symbols denote the initial locations of each agent and the goals of the other agent. In addition, we design two highly competitive games to further test the algorithms, namely, GW_nju (Fig. 2(e)) and GWa3 (Fig. 2(f)). The game GW_nju has two agents and the game GWa3 has three agents. The initial location and the goal of agent i are represented by the number i and G_i, respectively.

We first examine the performances of the tested algorithms regarding the SEE (steps of each episode) for each benchmark map (See Fig. 3). The state-of-the-art value function transfer mechanism helps NegoQ and ILs to converge fast in

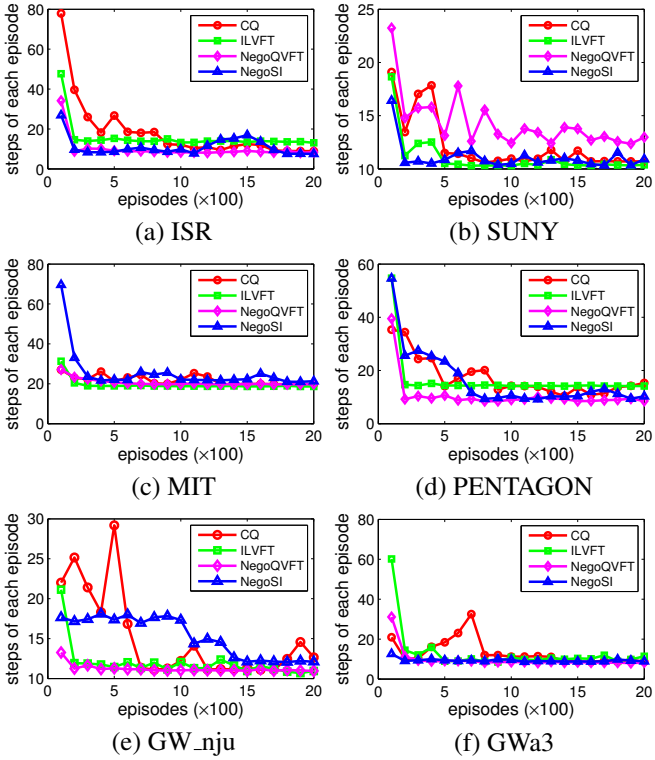

Figure 3: Steps of each episode for each tested benchmark map.

all games except for SUNY. NegoSI also has good convergence characteristics. CQ-learning, however, has a less stable learning curve especially in PENTAGON, GW_nju and GWa3. This is reasonable since in these highly competitive games, the agents' prior-learned optimal policies always have conflicts and CQ-learning cannot find the equilibrium joint action to coordinate them. When more collisions occur, the agents are frequently bounced to previous states and forced to take more steps before reaching their goals. In ISR, the learning step of final episode of NegoSI converges to 7.48, which is the closest one to the value obtained by the optimal policy. In other cases, NegoSI achieves the learning step of the final episode between 105.1% and 120.4% to the best solution, that is, the performance of NegoSI is on par with state-of-the-art methods, even without knowing all the joint state-action information.

The results regarding AR (average runtime) are shown in Table 1. Unsurprisingly, ILVFT has the fastest learning speed. CQ-learning only considers joint states in "coordination state" and it also has a relatively small computational complexity. The learning speed of NegoSI is slower than CQ-learning but faster than NegoVFT. Even if NegoSI adopts the sparse-interaction based learning framework and has computational complexity similar to CQ-learning, it needs to search for the equilibrium joint action in some states, which slows down the learning process .

4.2 A real-world application: intelligent warehouse systems

MARL has been widely used in such simulated domains as grid worlds [Busoniu *et al.*, 2008], but few applications have

Table 1: Average runtime for each tested map.

	ISR	*SUNY*	*MIT*	*PENTAGON*	*GW_nju*	*GWa3*
$CQ-learning$	8.54	5.91	13.68	8.52	5.07	5.95
$ILVFT$	4.91	4.14	9.78	6.56	2.53	7.21
$NegoQVFT$	13.92	20.18	36.84	18.45	21.89	50.48
$NegoSI$	16.74	7.33	19.58	16.18	7.08	16.41

been found for realistic problems comparing to single-agent RL algorithms or MAS algorithms. In this paper, we apply the proposed NegoSI algorithm to an intelligent warehouse problem. Previously, single agent path planning methods have been successfully used in complex systems [Zhou *et al.*, 2014], however, the intelligent warehouse employs a team of mobile robots to transport objects and single-agent path planning methods frequently cause collisions. So we solve the multi-agent coordination problem in a learning way.

The intelligent warehouse is made up of three parts: picking stations, robots and storage shelves (shown as in Fig. 4). There are generally four steps of the order fulfillment process for intelligent warehouse systems: 1) input and decompose orders into separated tasks; 2) task allocation; 3) path planning; 4) transportation and collision avoidance.

We focus on solving the multi-robot path planning and collision avoidance in a MARL way by requiring each robot to finish a certain number of random tasks. The simulation platform of an intelligent warehouse system is shown as in Fig. 4, which is a 16×21 grid environment and each grid is of size $0.5m \times 0.5m$ in real world. Shaded grids are storage shelves which cannot be passed through. The state space of each robot is made up of two parts: the location and the task number.

We only compare the NegoSI algorithm with CQ-learning in this problem. In fact, ILVFT has no guarantee of the convergence characteristics and it is difficult to converge in the intelligent warehouse setting. NegoVFT is also infeasible since its internal memory cost in MATLAB is estimated to be 4291 GB, while the costs of other algorithms are about 2 MB. Like NegoVFT, other MG-based MARL algorithms cannot solve the intelligent warehouse problem either. Experiments were performed in 2-agent and 3-agent settings, respectively, and the results are shown in Fig. 5 and Fig. 6.

In the 2-agent setting, the initial position and final goal of robot 1 are $(1,1)$ and those of robot 2 are $(1,16)$. Each robot needs to finish 30 randomly assigned tasks. The task set is the same for all algorithms. Robots with NegoSI achieve lower SEE than robots with CQ-learning throughout the whole learning process (as shown in Fig. 5). NegoSI finally converges to 449.9 steps and CQ-learning converges to 456.9 steps. In addition, the average runtime for completing all the tasks is 2227s for NegoSI and is 3606s for CQ-learning. In the 3-agent setting, the initial position and the final goal of different robots are $(1,1)$, $(1,8)$ and $(1,16)$. Each robot needs to finish 10 randomly assigned tasks. The task set is the same for different algorithms. SEE for robots with NegoSI finally converges to 168.7 steps and that for robots with CQ-learning converges to 177.3 steps (as shown in Fig. 6). In addition, the learning curves of NegoSI are more stable. The average runtime for completing all the tasks is 1352s for NegoSI and 2814s for CQ-learning.

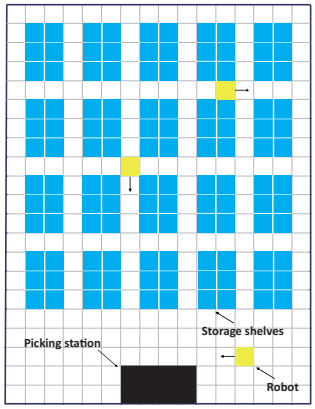

Figure 4: Simulation platform of the intelligent warehouse system.

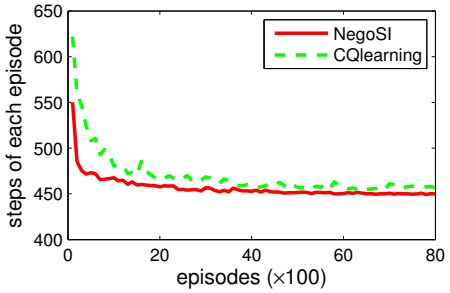

Figure 5: SEE (steps of each episode) for NegoSI and CQ-learning in the 2-agent intelligent warehouse.

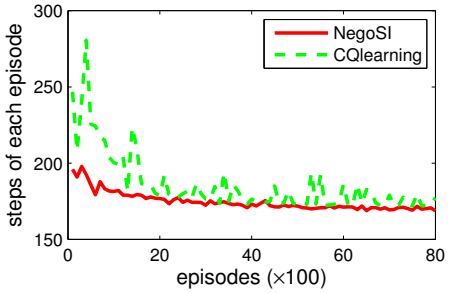

Figure 6: SEE (steps of each episode) for NegoSI and CQ-learning in the 3-agent intelligent warehouse.

5 Conclusions

In this paper a negotiation-based MARL algorithm with sparse interactions (NegoSI) is proposed for multi-agent coordination. The experimental results demonstrate the effectiveness of the presented NegoSI algorithm regarding such characteristics as fast convergence, low computational complexity and high scalability in comparison to the state-of-the-art MARL algorithms. Our future work focuses on more applications of MARL algorithms to practical problems.

References

[An *et al.*, 2011] Bo An, Victor Lesser, David Westbrook, and Michael Zink. Agent-mediated multi-step optimization for resource allocation in distributed sensor networks. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 609–616. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[Busoniu *et al.*, 2008] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):156–172, 2008.

[Claus and Boutilier, 1998] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pages 746–752, 1998.

[De Hauwere *et al.*, 2010] Yann-Michaël De Hauwere, Peter Vrancx, and Ann Nowé. Learning multi-agent state space representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 715–722. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[Greenwald *et al.*, 2003] Amy Greenwald, Keith Hall, and Roberto Serrano. Correlated q-learning. In *ICML*, volume 3, pages 242–249, 2003.

[Guestrin *et al.*, 2002] Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *ICML*, volume 2, pages 227–234, 2002.

[Hu and Wellman, 2003] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *The Journal of Machine Learning Research*, 4:1039–1069, 2003.

[Hu *et al.*, 2015a] Yujing Hu, Yang Gao, and Bo An. Learning in multi-agent systems with sparse interactions by knowledge transfer and game abstraction. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 753–761. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[Hu *et al.*, 2015b] Yujing Hu, Yang Gao, and Bo An. Multiagent reinforcement learning with unshared value functions. *Cybernetics, IEEE Transactions on*, 45(4):647–662, 2015.

[Kok and Vlassis, 2004] Jelle R Kok and Nikos Vlassis. Sparse cooperative q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 61. ACM, 2004.

[Littman, 1994] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157, pages 157–163, 1994.

[Littman, 2001] Michael L Littman. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pages 322–328, 2001.

[Melo and Veloso, 2009] Francisco S Melo and Manuela Veloso. Learning of coordination: Exploiting sparse interactions in multiagent systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 773–780. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[Melo and Veloso, 2011] Francisco S Melo and Manuela Veloso. Decentralized mdps with sparse interactions. *Artificial Intelligence*, 175(11):1757–1789, 2011.

[Nash and others, 1950] John F Nash et al. Equilibrium points in n-person games. *Proc. Nat. Acad. Sci. USA*, 36(1):48–49, 1950.

[Nowé *et al.*, 2012] Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. Game theory and multi-agent reinforcement learning. In *Reinforcement Learning*, pages 441–470. Springer, 2012.

[Stone and Veloso, 2000] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[Vrancx *et al.*, 2011] Peter Vrancx, Yann-Michaël De Hauwere, and Ann Nowé. Transfer learning for multi-agent coordination. In *ICAART (2)*, pages 263–272, 2011.

[Yu *et al.*, 2015] Chao Yu, Minjie Zhang, Fenghui Ren, and Guozhen Tan. Multiagent learning of coordination in loosely coupled multiagent systems. *Cybernetics, IEEE Transactions on*, 45(12):2853–2867, 2015.

[Zhou *et al.*, 2014] Luowei Zhou, Yuanyuan Shi, Jiangliu Wang, and Pei Yang. A balanced heuristic mechanism for multirobot task allocation of intelligent warehouses. *Mathematical Problems in Engineering*, 2014, 2014.

[Zhou *et al.*, 2016] Luowei Zhou, Pei Yang, Chunlin Chen, and Yang Gao. Multi-agent reinforcement learning with sparse interactions by negotiation and knowledge transfer. *IEEE Transactions on Cybernetics, preprint online, Doi: 10.1109/TCYB.2016.2543238*, 2016.