Multiagent Learning of Coordination in Loosely Coupled Multiagent Systems

Chao Yu, Minjie Zhang, Senior Member, IEEE, Fenghui Ren, and Guozhen Tan

Abstract—Multiagent learning (MAL) is a promising technique for agents to learn efficient coordinated behaviors in multiagent systems (MASs). In MAL, concurrent multiple distributed learning processes can make the learning environment nonstationary for each individual learner. Developing an efficient learning approach to coordinate agents’ behaviors in this dynamic environment is a difficult problem, especially when agents do not know the domain structure and have only local observability of the environment. In this paper, a coordinated MAL approach is proposed to enable agents to learn efficient coordinated behaviors by exploiting agent independence in loosely coupled MASs. The main feature of the proposed approach is to explicitly quantify and dynamically adapt agent independence during learning so that agents can make a trade-off between a single-agent learning process and a coordinated learning process for an efficient decision making. The proposed approach is employed to solve two-robot navigation problems in different scales of domains. Experimental results show that agents using the proposed approach can learn to act in concert or independently in different areas of the environment, which results in great computational savings and near optimal performance.

Index Terms—Agent independence, coordination, multiagent learning (MAL), reinforcement learning (RL), sparse interactions.

I. INTRODUCTION

MULTIAGENT learning (MAL) provides a promising paradigm for studying how agents can learn coordinated behaviors in multiagent systems (MASs) [1], and it is finding increasing applications in a wide range of real-world domains such as robotics [2], [3], distributed control [4], [5], resource management [6], [7] and automated trading [8], [9]. MAL uses techniques and concepts from disciplines such as artificial intelligence, game theory, psychology, cognition and sociology, and has attracted a great deal of interest in the research community in recent years [10]–[14]. The main challenge in MAL is that each learner must adapt its behavior concurrently in the context of other co-learners. This co-adaptation makes the learning environment nonstationary for each learner. Dynamics in such a nonstationary environment can cause the learning goal of a learner to change continuously, making MAL into a “moving-target learning” problem [1]. Moreover, agents in MAL need to use incomplete information to reason about other agents’ possible actions in order to make a reasonable decision.

MAL of coordination has been studied intensively. The need for coordination arises because the effect of an agent’s action on the environment also depends on the actions of other agents in the same environment. Hence, the agents’ actions must be mutually consistent in order to achieve their intended effects. This paper focuses on how to exploit inherent agent independent relationships for efficient CL, without imposing on the agents the important assumptions that are required in most existing approaches. This research is driven by the fact that many real-world MASs can exhibit a large amount of context-specific independence so that a general decision-making problem can be decomposed efficiently into sub-problems which are easier to solve [15]. In this kind of MASs, different levels of independence between agents can confine coordinated behaviors to some specific parts of the environment. An agent thus only needs to consider other agents’ information for coordination when it is necessary.

Many studies have investigated MAL of coordination by exploiting agent independence under different emphasis and assumptions. For example, hierarchical MAL approaches have been developed to take advantage of agents’ structural dependence to speed up the skill acquisition in cooperative MASs [16]–[18]. Agent independence in these approaches is represented by a decomposition of a main task into several subtasks, each of which is relevant to some particular agents. Other examples include coordination-graph based learning approaches [19], [20] and the distributed value function approach [21], with the focus on studying how an optimal global joint policy can be achieved through local interactions among agents in a networked interaction structure. All these approaches focus on coordinated MAL when an explicit representation of agent independence is given beforehand. This explicit representation is indicated either by a predefined task decomposition or by a fixed interaction structure. In many cases, however, agents must learn coordinated behaviors when an explicit representation of agent independence is not available as a prior knowledge. As a result, agents need to build up such a representation from their own learning
experiences. Some approaches to coordinated MAL, therefore, have been proposed to enable agents to learn their dependent relationships so that coordination is only considered when agents are really dependent on each other [20], [22]–[24]. These approaches, however, are based on prior knowledge of each agent’s individual optimal policy [23] or on the assumptions of agents’ full observability of the joint-state and/or joint-action of other agents [20], [22], [24]. These preconditions in existing approaches put great limits to the development of efficient coordinated MAL approaches in practical MASs.

Against this background, this paper proposes an approach to exploit agent independence for efficient CL, without requiring agents’ knowledge about the location of coordination or full observability of the environment. We firstly formalize agent independence with the aim of decomposing the decision-making process into sub-processes more efficiently than that in general decision-making models of MAL. We then propose a coordinated MAL approach which enables agents to learn a trade-off between a single agent learning process and a CL process through dynamic adaptation of the independence between agents. The single agent learning process can take advantage of the agents’ independence from one another in order to achieve computational savings, while the CL process can take advantage of each agent’s limited observability to mitigate the uncertainties of the learning environment in order to improve learning. Through the dynamic trade-off between these two processes, agents can learn an efficient coordinated policy with minimum computational consumption. The proposed approach is employed to solve different scales of robot navigation problems. Experimental results show that agents using our approach can learn to act in concert or independently in different regions of the environment according to the approximated independence between the agents. A large amount of computational savings can be achieved and at the same time a near optimal performance can be guaranteed by using the proposed approach.

The rest of this paper is organized as follows. Section II introduces general decision-making models in MAL. Section III gives the description of robot navigation problems and formal definition of agent independence. Section IV describes the proposed MAL approach and Section V presents the experimental results. Section VI compares this paper with related studies and finally Section VII draws some conclusions and suggests directions for future research.

II. GENERAL DECISION-MAKING MODELS IN MAL

This section briefly reviews the single-agent Markov decision process (MDP) model and the extended multiagent model. Fundamental concepts are clarified and notations are established for further description.

A. MDP

A MDP can be used to describe a single-agent sequential decision-making problem in which an agent must choose an action at every time step to maximize some reward-based functions [25]. Formally, an MDP can be defined by a four-tuple \( M = (S, A, P, R) \), where \( S \) is a finite state space, \( A \) is a set of actions available to the agent, \( P(s,a,s') : S \times A \times S \rightarrow [0,1] \) is a Markovian transition function when the agent transits from state \( s \) to \( s' \) after taking action \( a \), and \( R : S \times A \rightarrow R \) is a reward function that returns immediate reward \( R(s,a) \) to the agent after taking action \( a \) in state \( s \). An agent’s policy \( \pi : S \times A \rightarrow [0,1] \) is a probability distribution that maps a state \( s \in S \) to an action \( a \in A \).

Fig. 1 gives an example of MDP regarding running a company. In every state of the company (famous or unknown and rich or poor), the company has two actions of either saving money (\( S \)) or advertising (\( A \)). The company will transit to another state according to certain probability using either action and receive a reward (e.g., +10 in state rich and unknown). The goal in an MDP is to learn a policy \( \pi \) so as to maximize the expected discounted reward \( V^\pi(s) \) for each state \( s \in S \). The reward function is given by

\[
V^\pi(s) = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) | s_0 = s \right] \tag{1}
\]

where \( E_\pi \) is the expectation of policy \( \pi \), \( s_t \) denotes the state at time \( t \), and \( \gamma \in [0,1) \) is a discount factor.

For any finite MDP, there is at least one optimal policy \( \pi^* \), such that \( V^{\pi^*}(s) \geq V^\pi(s) \) for every policy \( \pi \) and every state \( s \in S \). \( \pi^* \) can be computed by using linear programming or dynamic programming techniques if an agent fully knows the reward and transition functions of the environment. When these functions are unknown to the agent, finding an optimal policy in MDP can be solved using reinforcement learning (RL) [26] methods, in which an agent learns through trial-and-error interactions with its environment. One of the most important and widely used RL approach is Q-learning [27], which is an off-policy model-free temporal difference (TD) control algorithm. In Q-learning, an agent makes a decision through the estimation of a set of Q values. The one step updating rule is given by (2), where \( \alpha \in (0,1] \) is a learning rate

\[
Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t \left[ R(s,a) + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a) \right]. \tag{2}
\]
Every Q value of a state-action pair can be stored in a table for a discrete state-action space. It is proved that this tabular Q-learning method converges to the optimal $Q^*(s, a)$ w.p.1 when all state-action pairs are visited infinitely and an appropriate exploration strategy and learning rate are chosen [27].

**B. Decentralized MDPs (Dec-MDPs)**

A Dec-MDPs model is an extension of the aforementioned single agent MDP model to allow decentralized decision makings of multiple agents [28]. In Dec-MDPs, at each step, each agent has a local observation and subsequently chooses an action. An agent’s state transitions and rewards depend on the actions of all the agents. More formally, a Dec-MDPs model can be defined by a tuple $N = (n, S, \{A_i\}, P(s, a, s'), R(s, a))$, where $n$ is the number of agents; $S$ is a finite set of joint states of all agents; $A_i(i \in \{1, n\})$ is a finite set of actions available to agent $i$; $P(s, a, s')$, where $s \in S$ is the joint state of all the agents and $a \in A (A = \times_{i=1}^{n}A_i)$ is the joint action of all the agents, represents the transition probability from state $s$ to state $s'$ when the joint action $a$ is taken in state $s$; and $R(s, a)$ represents the reward received by the agents when joint action $a$ is taken in state $s$. For simplicity, we write $a_i = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ to denote the reduced action of agent $i$, thus the joint action $a = (a_1, \ldots, a_n)$ can be simply represented as $a = \langle a_1, a_n \rangle$. This paper focuses on a more specialized version of the Dec-MDPs model, that is, factored Dec-MDPs [15], [29], in which the system state can be factored into $n + 1$ distinct individual components so that $S = S_0 \times S_1 \times \cdots \times S_n$, where $S_0$ denotes an agent-independent component of the state (i.e., assumed common knowledge among all agents), $S_i(i \in \{1, n\})$ is the state space of agent $i$. For each agent, its local state $s_i$ is defined as $s_i = \langle s_0, s_i \rangle (s_i \in S_i \times S_0)$, where $s_j \in S_j$ represents the portion of global state specific to agent $i$ and $s_0 \in S_0$ represents the portion of the global state shared among all agents.

The Dec-MDPs model is a particular case of the decentralized partial observable MDP model [30], where every agent has a partial observability so that the agent cannot determine its local state unambiguously through its local observation. In Dec-MDPs, however, an agent can have a local observability, which means that each agent can determine the corresponding local state from its local observations. In other words, agents in Dec-MDPs, altogether, have a joint full observability, or collective observability [31], which means that each agent in Dec-MDPs observes a part of the state and the combined observations of all agents can uniquely identify the overall state. This is in contrast with multiagent MDPs [32], [33], where each agent, individually, already has full observability allowing the individual observation of an agent to identify the overall state unambiguously. Due to agents’ local observability of the environment, each agent in Dec-MDPs cannot access the global state/reward information to make a reasonable decision. The decentralized decision makings in Dec-MDPs thus bear an extremely high complexity (i.e., NEXP-complete) to achieve a globally optimal policy, even in the two-agent case, versus the P-completeness of a single MDP and multiagent MDPs [28].

**III. PROBLEM DESCRIPTION AND DEFINITIONS**

This section first gives a brief description of the robot navigation problems, which are a representative domain for loosely coupled MASs. The concept of agent independence is then formally defined for the purpose of inducing the proposed approach.

**A. Robots Navigation Problems**

Robot navigation problems (or referred to as grid world puzzles) are fundamental research issues in the AI community. Fig. 2 shows three domains of robot navigation problems. In each of these domains, two robots $R_1$ and $R_2$ are navigating in an environment, each trying to reach its own goal $G_1$ and $G_2$, respectively (in Fig. 2(b), $G_1$ and $G_2$ are in the same grid denoted by $G$). Each robot has four actions, i.e., “Move East,” “Move South,” “Move West,” and “Move North.” The action space is therefore represented as all the joint actions of both robots. Each robot’s state is its location in the environment. The state space is therefore represented as all possible joint locations of both robots. The decision-making process can be modeled as Dec-MDPs, in which each robot has to make a decision independently to achieve an optimal global policy through its local observation (individual action and state). Each robot can only have a local observability of the environment, which means the robot cannot observe the other robot’s location to determine the overall state of the Dec-MDPs. Furthermore, both robots are coupled with a shared global transition and reward function, but neither robot can access such global transition and reward functions on its own. A robot thus cannot explicitly determine the effect of its individual decision-making on the environment in order to take a reasonable action.

As can be seen from Fig. 2, interactions between the robots do not occur very often in most states of the environment. This means that generally each robot can make its own decision without regard to the other robot’s state and/or action but in certain specific situations (e.g., near the doorway), the robots are tightly coupled, and thus need to coordinate with each other for better performance. The sparse interactions between the robots can cause varying levels of independence in the states so that coordinated behavior is only heavily required in some local parts of the environment. Intuitively, the robots have higher levels of independence in the states that are further away from the doorway/entrance than those in the states nearer to the doorway/entrance. This is because that robots are more prone to have collisions near the doorway/entrance so that a robot’s transition function and/or reward function

![Fig. 2. Three small-scale robot navigation domains. (a) HG. (b) TTG. (c) TR.](image-url)
may heavily depend on the other robot. By exploiting the independence between robots, it is thus possible to decompose the general decision-making process into two distinct sub-processes. One is to let each robot make its own decision individually and completely disregard the existence of the other robot when there is an independence between robots. The other is to let robots coordinate their behaviors with each other by considering the situation of the other robot when interactions (in terms of coordination) are inevitable. Through this decomposition, the original problem can be more easily solved as the uncertainties caused by incomplete information can be greatly reduced.

Many approaches have been proposed to solve robot navigation problems by exploiting the independence of robots [22], [23], [34]–[36]. These approaches simplify the problem by predefining the dangerous areas for the robots [34], [35], or by imposing some assumptions on each robot, either its individual optimal policy [23], or the full observability of the other robot’s state [22] and action [36]. This paper, however, proposes an approach which enables the robots to devise an efficient coordinated policy without prior knowledge of either the domain structure or the robot itself. Before introducing this approach, it is necessary to give a formal definition of agent independence in order to clarify how a general Dec-MDPs problem can be decomposed efficiently by exploiting agent independence.

B. Formalization of Agent Independence in MASs

As can be seen in Fig. 2, the agents (robots) can have certain levels of independence (in terms of state transitions and rewards) so as to allow a lower computational complexity for solving the Dec-MDPs problem. Much work has been done to exploit these kinds of independence between agents in order to achieve an efficient decision-making in Dec-MDPs. In [37], a specific Dec-MDPs model called transition-independent Dec-MDPs [37], [38] is identified, in which the overall transition function \( P \) can be separated into \( n \) distinct functions \( P_0, \ldots, P_n \) as \( P = P_0 \prod_{i=1}^n P_i \), where each \( P_i(i \in [1, n]) \) stands for an individual transition function of agent \( i \) and \( P_0 \) is an initial transition component. For any next state \( s'_i \in S_i \) of agent \( i \), \( P_i \) is given by \( P_i(s, a, s') = P_i(s_i, a_i, s'_i) \). In other words, the next local state of each agent is independent of the local states of all other agents, given this agent’s previous local state, local action, and the external system component (\( S_h \)). It has been shown that the decision-making complexity in transition-independent Dec-MDPs is NP-complete, which is a significant reduction compared to that of general Dec-MDPs models [38]. It is argued, however, that not all multiagent domains are fully transition-independent [15]. Furthermore, due to the shared reward component, it is still a nontrivial task to solve transition-independent Dec-MDPs.

Similarly, a Dec-MDPs model can also be reward-independent where the joint reward function \( R \) is represented as a function of individual reward functions \( R_1, \ldots, R_n \) by \( R(s, a) = f(R_1(s_1, a_1), \ldots, R_n(s_n, a_n)) \) [5], [29]. It was recently shown that a reward-independent Dec-MDPs problem retains NEXP-complete complexity [29], [39]. However, when associated with transition independence, reward independence implies that a Dec-MDPs model can be decomposed into \( n \) independent MDPs, each of which can be solved separately. The complexity of reward-independent Dec-MDPs thus becomes the same P-complete as standard MDPs.

General transition/reward-independent Dec-MDPs decompose the overall function \( P/R \) into each agent’s individual function \( P/R_i \), which depends only on each agent’s local state \( s_i \) and individual action \( a_i \). Since an agent is potentially affected by other agents, the agent’s local state and individual action usually cannot fully determine its individual function \( P/R_i \). Thus, in [35], an agent’s individual function is defined on the parameters of the overall state and action of all the agents. The individual function can then be decomposed into a local individual component based on the agent’s local information \( s_i \) and \( a_i \), and an interaction component based on all the agents’ information \( s \) and \( a \). The factorization in [35] is based on agents’ prior knowledge about the domain structure in terms of the location of interaction states where agents mutually affect each other’s transitions/rewards. Outside these interaction states, a full independence is assumed so that all the agents can make decisions individually. In other words, a state is identified as one in which the agents are either completely independent or completely dependent on each other. This handling of independence is always intractable due to the complexity of Dec-MDPs and the lack of prior knowledge about the domain structure. To better reflect the uncertainties of agent independence, the transition/reward-independent degree is introduced in the factorization of individual functions as given by Definition 1.

Definition 1 (Transition/Reward-Independent Degree):

\[ P_k \left( \langle s_k, s_{-k} \rangle, \langle a_k, a_{-k} \rangle, \langle s'_k, s'_{-k} \rangle \right) = \frac{P}{\mu k} \left( \langle s_k, a_k, s'_k \rangle \right) + (1 - \mu k) P_d \left( \langle s_k, s_{-k}, a_k, a_{-k} \rangle, \langle s'_k, s'_{-k} \rangle \right) \]

\[ R_k \left( \langle s_k, s_{-k} \rangle, \langle a_k, a_{-k} \rangle \right) = \frac{R}{\nu k} \left( \langle s_k, a_k \rangle \right) + (1 - \nu) R_d \left( \langle s_k, s_{-k}, a_k, a_{-k} \rangle \right) \]

In Definition 1, \( P^k(\langle s_k, a_k, s'_k \rangle) \) is agent \( k \)’s local individual transition function that depends only on agent \( k \)’s local state and action, and \( P^k(\langle s_k, s_{-k}, a_k, a_{-k} \rangle, \langle s'_k, s'_{-k} \rangle) \) is agent \( k \)’s interactive transition function that depends on the overall information of all agents. \( R^k(\langle s_k, a_k \rangle) \) and \( R^k(\langle s_k, s_{-k}, a_k, a_{-k} \rangle) \) are agent \( k \)’s local individual reward function and interactive reward function, respectively.

Transition/reward-independent degree \( \mu/\nu \) signifies the uncertainties of independence between agent \( k \) and the remaining agents in terms of transition/reward function. It is assumed that if \( P^k > P^d \), then \( R^k > R_d \) holds, and vice versa. This means that if an agent is more likely to determine its transition function through its local information, then the agent is also more likely to determine its reward function through its local information. Agent \( k \)’s overall independence with the
remaining agents, however, is signified by the value function $Q$ [see (1)]. The independence of agents can thus be formally defined by Definition 2.

Definition 2 (Independent Degree): An independent degree $\xi \in [\min\{\mu, v\}, \max\{\mu, v\}]$ is a value that signifies the extent of independence of agent $k$ regarding to the remaining agents, so that the individual value function can be decomposed according to (5), where $Q^k$ and $Q^\ell$ are the expected local and interaction value function of agent $k$, respectively

$$Q_k((s_k, s_{-k}), (a_k, a_{-k})) = \xi Q^k(\hat{s}_k, a_k) + (1 - \xi) Q^\ell((\hat{s}_k, s_{-k}), (a_k, a_{-k})). \tag{5}$$

Lemma 1: The independent degree $\xi$ is well-defined, that is, it always exists and is unique.

Proof: Let $a = (a_k, a_{-k}), s = (s_k, s_{-k})$. For each value function of agent $a_k$, $Q_k(a, s)$ depends on agent $a_k$’s transition function $P_k$ and reward function $R_k$. The other agents’ influence on agent $a_k$ is embodied in the common interaction components of transition function $P^\ell$ and reward function $R^\ell$. Thus

$$Q_k(a, s) = R_k(a, s) + \gamma \sum_{s'} P_k(a, s', b)\max_{a'} Q_k(b) = R_k(\hat{s}_k, a_k) + \gamma \sum_{s'} [\mu P^k(\hat{s}_k, a_k, s_k') + (1 - \mu) P^\ell(s_k, s_{-k}, s')] \max_{a'} Q_k(s', a')$$

\[= R_k(\hat{s}_k, a_k) + \gamma \sum_{s'} P^k(\hat{s}_k, a_k, s_k') \max_{a'} Q_k(s', a') + (1 - \mu) \gamma \sum_{s'} P^\ell(s_k, s_{-k}, s') \max_{a'} Q_k(s', a') \leq R_k(\hat{s}_k, a_k) + (1 - \mu) \gamma \sum_{s'} P^\ell(s_k, s_{-k}, s') \max_{a'} Q_k(s', a') \leq R_k(\hat{s}_k, a_k) + \gamma \sum_{s'} P^\ell(s_k, s_{-k}, s') \max_{a'} Q_k(s', a') \]

Combining (6) and (8) can have

$$Q_k(s, a) = \xi Q^k(\hat{s}_k, a_k) + (1 - \xi) Q^\ell(s, a). \tag{10}$$

Note that, for all agents, if $\xi = 1$ holds in every state, that is, both transition and reward independence are completely achieved for these agents, a Dec-MDPs model can be reduced to a set of independent MDP subproblems, each of which can be solved separately. The complexity of this class of models can thus become the same P-complete as standard MDPs. In other situations, the Dec-MDPs problem becomes an NEXP-complete puzzle, even for the simplest two-agent scenario [28]. One straightforward way to reduce the complexity is to provide the agents with sufficient information to overcome the uncertainties caused by agents’ local observability. This means allowing an agent to observe other agents’ information for decision-making, either through full observability of the environment or unlimited communication capability. Keeping all the other agents’ information during learning, however, becomes intractable as the search space grows exponentially with the number of agents, and communication/observability is always restricted in real-life applications. A direct solution to avoid this dilemma is to let agents learn to use other agents’ information for coordination only when necessary. The independent degrees, which capture the different levels of independence in Dec-MDPs, can signify the extent of such a necessity, and can thus be exploited for more efficient decision-making without requiring an agent to consider all information from other agents, most of which is redundant as indicated by the context-specific independence.

IV. COORDINATED MAL APPROACH

A coordinated MAL approach is proposed through dynamic adaptation of the independent degrees during the learning process so that coordination can be achieved with low computational complexity. This approach estimates the optimal global policy for each agent when the model is not well specified. A model is not well specified when the transition/reward functions as well as independent degrees are unknown to the agents. Note that, although the transition and reward functions can usually be defined beforehand for a Dec-MDPs model, an explicit specification of independent degrees is not straightforward due to the complexities and
A. Principle of the Learning Approach

As illustrated above, a hidden independent degree always exists in a state of a Dec-MDPs model, signifying the necessity of coordination between agents. A higher independent degree means that an agent is more independent from the remaining agents and can learn to build a model of the environment with higher certainty using only its individual information. In this situation, an agent can thus conduct an independent learning (IL) process by disregarding the existence of other agents. More specifically, when agent \( i \) has an independent degree of \( \xi^k_i \) in state \( s^t_i \), it will apply IL with a probability of \( \xi^k_i \). Otherwise CL can be conducted to combine other agents’ information for decision-making with a probability of \( 1 - \xi^k_i \). The CL process can be carried out by assuming that each agent has limited observability of the environment, that is, distance \( (s_1, s_{-i}) \leq R : P[S(t) = s_{O_1}(t) = o_1] = 1 \), where \( S(t) = (s_1, s_{-i}) \) is the joint state of all agents, \( o_1 \) is the individual observation of agent \( i \), and \( R \) is the agent’s perception distance. This means that only when other agents are in the perception distance of agent \( i \) (i.e., distance \( (s_1, s_{-i}) \leq R \)) can agent \( i \) recover the joint state unambiguously through its individual observation. Equipping each agent with limited observability of the environment is reasonable because in many real-life applications agents can observe each other only when they are spatially close [35]. Here, the CL process is assumed to impose no cost on the agents, that is, whether or not to coordinate with other agents is completely determined by the value of independent degrees and this process has no probability to fail. The fact that an agent always succeed in activating the CL process, however, does not mean that the agent can successfully receive other agents’ information all the time since agents are only endowed with limited observability of the environment. An agent starts with an initial belief that there is no dependent relationship with the other agents. The agent then adapts the independent degrees dynamically during learning to achieve an efficient coordinated policy. This policy can make a trade-off between the IL process and the CL process to achieve computational savings and to mitigate the uncertainties of the learning environment. Through the dynamic trade-off between these two processes, agents can achieve efficient learning at the expense of minimum computational consumption. The sketch of this learning approach is given by Algorithm 1, where line 4 indicates the dynamic trade-off process, which is determined by the explicit independent degrees, and lines 5–8 show how to adapt the independent degrees to best reflect the real independence of the agents.

As agents are learning through trial-and-error interactions with the environment, the only available information to an agent is the agent’s learning experience and the immediate rewards from the environment. When a conflict occurs (i.e., penalized reward is received) in a state, the agent recognizes that its decision may depend on other agents’ decisions as well (i.e., coordination might be required) in the corresponding state. The estimated independence thus can be updated by considering both the potential contributions from other domain states and the most direct causes of the conflict based on the agents historical experiences.

More specifically, a penalized reward received in state \( s \) is implicitly contributed to by all other states in the domain, with the extent of the effect determined by the difference between other domain states and the most direct causes of the conflict. When a reward \( r \) is achieved in state \( s^* \), \( f^s_{\pi^*}(s) \) has the highest value signifying that reward \( r \) is mainly caused by state \( s^* \), and those states having the same similarity with the conflicting state \( s^* \) are assumed to play the same role of causing the reward in state \( s^* \). As the similarity between \( s \) and \( s^* \) decreases, state \( s \) has a lower effect on causing reward \( r \), which is reflected by the lower value of \( f^s_{\pi^*}(s) \).

**Algorithm 1:** General Learning Approach for Agent \( i \)

1. Initializes learning parameters;
2. For all state \( s^t_i \in S_i \), \( \xi^k_i \leftarrow 1 \);
3. if agent \( i \) is in state \( s^t_i \) then
4. Conducts independent learning with a probability of \( \xi^k_i \) or conducts coordinated learning with a probability of \( 1 - \xi^k_i \);
5. Receives penalized reward \( r_i \) and search for causes of the reward;
6. for each state \( s^t_i \in S_i \) do
7. Calculates similarity \( \zeta(s^t_i, s^t_j) \) and updates eligibility trace \( e^t_i \);
8. Adjusts \( \xi^k_i \) according to \( \zeta(s^t_i, s^t_j) \) and \( e^t_i \) based on the diffusion function \( f^t_i(s^t_j) \);

In many cases, the roles of the states having the same similarity with a conflicting state can be different. For example, when an agent detects a conflict in state $s^k$, it wants to determine the roles of the neighboring states of state $s^k$, say $s_1$, $s_2$, $s_3$, and $s_4$, to cause this conflict. If the past statistical information indicates that the agent usually transits from states $s_1$ and $s_2$ to $s^k$, causing the conflict in state $s^k$, but in some cases it transits from state $s^k$ to states $s_3$ and $s_4$. It is obvious that states $s_3$ and $s_4$ are similar with states $s_1$ and $s_2$, but are not the causes of the conflict in $s^k$. Thus, agents should assign credit to those states that are really responsible for the resulting conflict. To solve this problem, eligibility trace (ET) is introduced in order to make a temporary record of the occurrence of an event, i.e., a penalized reward is received. ET is one of the basic mechanisms of RL to speed up the learning process. For example, in TD($\lambda$) algorithms [26], $\lambda$ refers to the use of an ET. Unlike TD($\lambda$), where ET is updated at every step during learning to boost learning efficiency, in the proposed approach, ET is updated only when an event (i.e., a conflict) occurs. Let $S^e$ be the state trajectory that is caused by an event and $\varepsilon^e_{t^i}$ be the ET value of agent $i$ in state $s^e_t$, then for each state $s^e_t \in S^e$, the ET value can be updated by (12), where $\gamma \in [0, 1)$ is a discount rate and $\lambda \in [0, 1]$ is the trace-decay parameter.

$$
\varepsilon^e_{t^i}(t + 1) = \begin{cases} 
\gamma^r \varepsilon^e_{t^i}(t) + 1, & s^e_t \in S^e \\
\gamma^\lambda \varepsilon^e_{t^i}(t), & \text{else},
\end{cases}
$$

(12)

Based on the diffusion function and ET, a penalized reward $r(t)$ signifying the necessity of coordination can be diffused to all the states that are potentially eligible for this reward. The independent degree can thus be adjusted

$$
\xi^k(t + 1) = G\left(\psi^k_{t}(t + 1)\right),
$$

(13)

$$
\psi^k_{t}(t + 1) = \psi^k_{t}(t) + \varepsilon^k_{t}(t)\phi^k_{t}(t)\left(s^e_t\right) \text{ where } \psi^k_{t}(0) = 0.
$$

(14)

In (13), $G(x)$ is a normalization function to map the value of $x$ to interval $[0, 1]$, with a lower value of $x$ corresponding to a higher value of $G(x)$. In (14), $\psi^k_{t}(t + 1)$ is a value to signify the necessity of coordination, $s^e(t)$ is the state resulting in the penalized reward $r(t)$, and $\varepsilon^k_{t}(t)$ is the ET value of agent $i$ in state $s^e_t$.

B. Explicit Learning Algorithm

An explicit Q-learning based algorithm derived from the above approach (refer to Algorithm 1) is proposed in Algorithm 2 to solve robot navigation problems from the perspective of agent (robot) $i$. Although other learning techniques, such as the on-policy TD control method Sarsa, Actor-Critic methods [26], are also suitable, Q-learning has been chosen as an illustration because of its robustness, relative simplicity and widely successful applications in a number of multiagent domains. This paper focuses on two-agent scenarios where agent $i$ and $j$ are learning concurrently in the same environment. Although two-agent scenarios greatly simplify general robot navigation problems, most current research into coordinated MAL in robot navigation problems still focuses on this setting because two-agent scenarios already encompass the main challenges encountered in general MAL problems, that is, concurrent learning dynamics, each agent’s local observability, and limited or no prior knowledge concerning the domain or the agents. The main contribution of this paper is thus the proposed idea that an explicit quantified representation of the agent independence can be built up for efficient CL, without requiring the assumptions which are needed in most existing approaches.

In Algorithm 2, $Q_i(s_t, a_t)$ is a single-state Q-value table for agent $i$. $Q_i(s_t, a_t)$ is a joint-state Q-value table for both agent $i$ and agent $j$. $\xi^k_{t}(t)$ and $\varepsilon^k_{t}(k = 1, \ldots, m)$ are the values of the independent degree and ET, respectively, when agent $i$ is in state $s^e_t$, and trajectory_list is a list to store the trajectory (i.e., the state transition history) of the agent. Agent $i$ decides whether to coordinate with agent $j$ based on the independent degree in a state (line 4). If agent $i$ chooses action coordinate (line 5), it will activate its perception process to determine the local state information of agent $j$ through its limited observability of the environment (lines 6–11). Otherwise, agent $i$ chooses its action based on single-state Q-value table $Q_i(s_t, a_t)$ (line 12). The perception process can be carried out either through agent $i$’s limited observing capability or by using explicit communication with $j$. In robot navigation problems, a robot can rely on its sensor to locate the other robot or send a message to require the other robot to divulge its location [34]. Agent $i$ in the perception process, however, does not necessarily succeed in accessing agent $j$’s state information as this success is environment-dependent, which means that only when agent $j$ is really located within the perception distance of agent $i$ can agent $i$ receive agent $j$’s state information (line 6).

---

**Algorithm 2: Coordinated Learning Algorithm for Agent $i$**

1. Initializes $Q_i(s_t, a_t)$, $Q_i(s_t, a_t) \leftarrow \emptyset$, $\xi^k_{t}(t) \leftarrow 1$, $\varepsilon^k_{t}(t) \leftarrow 0$, trajectory_list $\leftarrow \emptyset$
2. for each episode $e$ ($n=1, \ldots, E$)
   for each step $t$ ($t=1, \ldots, T$)
     Generates a random number $\tau$, $\tau \in [0, 1]$;
     if $\xi(s_t) \leq \tau$ then
       if agent $j$ is in vision then
         PerceptionFlag $\leftarrow$ True and read $s_j$, $j_s \leftarrow \langle s_j, j_t \rangle$
         if $j_s$ not in table $Q_i(j_t, a_t)$ then
           adds $j_s$ to table $Q_i(j_t, a_t)$;
           selects $a_t(j_t)$ from $Q_i(j_t, a_t)$;
         else selects $a_t(j_t)$ from $Q_i(s_t, a_t)$;
         trajectory_list.add($s_t$), transit to state $s_t'$ and receive $r_t(t)$;
       if $\xi(s_t) \leq \tau$ and PerceptionFlag $\leftarrow$ True then
         Updates $Q_i(j_t, a_t) \leftarrow Q_i(j_t, a_t) + \alpha [r_t(t) + \gamma \max_{a_t} Q_i(s_t', a_t') - Q_i(s_t, a_t)]$;
       else
         Updates $Q_i(s_t, a_t) \leftarrow Q_i(s_t, a_t) + \alpha [r_t(t) + \gamma \max_{a_t} Q_i(s_t', a_t') - Q_i(s_t, a_t)]$;
     Call Algorithm 3 to adjust $\xi^k_{t}(t)$, $s_t \leftarrow s_t'$;
After the successful perception process, agent \( i \) makes use of the local state information from agent \( j \) to choose its action based on the joint-state Q-value table \( Q_c(js_i, a_i) \) (lines 7–10). Otherwise, agent \( i \) makes its decision based only on single-state Q-value table \( Q_i(s_i, a_i) \) (line 11). After each transition (line 13), agent \( i \) updates its Q values (signifying how to coordinate with agent \( j \)) (lines 14–17) and independent degrees (signifying when coordination is beneficial) (line 18), respectively.

The dynamic adaptation process of independent degrees is given by Algorithm 3, where \( s_i^k(x_k, y_k) \) is a local state with coordinate \( (x_k, y_k) \), \( s_i(t) \) is the coordinate of the conflicting state \( s_i(t) \), and \( \psi \) is a value to signify the necessity of coordination [see (14)]. As the similarity between \( s_i^k \) and \( s_i(t) \) decreases (i.e., the Euclidean distance of these two states increases, \( \xi \)), the value of the diffusion function decreases, indicating that state \( s_i^k \) has a lower impact on the cause of the conflict in state \( s_i(t) \) (line 7).

After the successful perception process, agent \( i \) makes use of the local state information from agent \( j \) to choose its action based on the joint-state Q-value table \( Q_c(js_i, a_i) \) (lines 7–10). Otherwise, agent \( i \) makes its decision based only on single-state Q-value table \( Q_i(s_i, a_i) \) (line 11). After each transition (line 13), agent \( i \) updates its Q values (signifying how to coordinate with agent \( j \)) (lines 14–17) and independent degrees (signifying when coordination is beneficial) (line 18), respectively.

The dynamic adaptation process of independent degrees is given by Algorithm 3, where \( s_i^k(x_k, y_k) \) is a local state with coordinate \( (x_k, y_k) \), \( s_i(t) \) is the coordinate of the conflicting state \( s_i(t) \), and \( \psi \) is a value to signify the necessity of coordination [see (14)]. As the similarity between \( s_i^k \) and \( s_i(t) \) decreases (i.e., the Euclidean distance of these two states increases, \( \xi \)), the value of the diffusion function decreases, indicating that state \( s_i^k \) has a lower impact on the cause of the conflict in state \( s_i(t) \) (line 7).

The value of the diffusion function is combined with the eligible trace value \( \epsilon_i^k(t) \) to indicate which previous states are really responsible for the conflict in state \( s_i(t) \). As for the normalization function \( G(x) \) in (13), the maximum \( \psi \) (denoted as \( \text{max} \text{temp} \)) after each episode is chosen (line 8) to let the values of \( \psi \) in all other states be compared with \( \text{max} \text{temp} \) so as to confine \( \psi \) to [0, 1] (line 10). This way of handling the normalization function \( G(x) \) is used as an illustration here, but a number of other methods can be adopted to define a concrete normalization function.

Several discussions are laid out as follows.

1) As can be seen from Algorithm 2 (line 15), the update of joint-state Q-value \( Q_c(js_i, a_i) \) uses the estimates of individual single-state Q-value \( Q_i(s_i, a_i) \) in the next step. The relation between individual single-state Q-value \( Q_i(s_i, a_i) \) and the joint-state Q-value \( Q_c(js_i, a_i) \) is illustrated in Fig. 3. An agent chooses its action from individual Q-value \( Q_i(s_i, a_i) \) and updates this value by using the maximum estimated Q-value of the state in next step (say, state 7) with a probability of \( \xi \). Otherwise, the agent chooses an action from joint-state Q-value \( Q_c(js_i, a_i) \) with a probability of \( 1 - \xi \) and updates the joint-state Q-value (say, joint state 4-1) by using the maximum individual Q-value \( Q_i(s_i, a_i) \) in the next step. From Fig. 3, we can see that the joint-state Q-value of different state/actions in table \( Q_c(js, a) \) are independent from each other, and the joint-state Q-values \( Q_c(js_i, a_i) \) only determines one-step coordinated behavior of an agent. This implies that if agents are loosely coupled (as in robot navigation problems) and conflicts between agents are only confined to some particular parts (usually accounting for a small proportion) of the environment, the joint-state Q-value table can be very sparse. Therefore, learning for an optimal global policy in this situation will be computationally inexpensive (almost the same with directly learning with individual single-state Q-values).

2) In the robot navigation problem, there are multiple optimal policies to avoid an uncoordinated joint action in the conflicting states. For example, when both robots come to the doorway in the HG domain, either robot choosing to pass the doorway while the other makes a detour can constitute an optimal joint action. Both robots should decide separately which joint action to choose to avoid mis-coordination. This type of difficulty is called an “equilibrium selection problem” and it is inherent in any CL process. In the proposed approach, the equilibrium selection process is carried out by using only the agents’ joint-state information. This is in contrast with some other approaches which use joint-action information [40], or joint-state-action information [36], or simply a random action selection mechanism [22], to conduct the CL process. It is obvious that learning from the joint-action information of both agents can lead to a coordinated joint action. However, as shown by the experimental results, learning directly from the joint-state information of both agents can also produce such a coordinated joint action. Importantly, this implies that joint-state information alone can be enough for agents to learn coordinated behaviors. Learning directly from the joint-action information or requiring further such information will thus be redundant in an equilibrium selection process.

3) In the proposed approach, agents adapt their independence every time an event (a conflict) occurs. This means the approach is fully event-driven, and might be potentially applied to other similar domains as long as a significant conflicting event can happen due to
the dependence of the agents. Furthermore, an agent updates its Q values (signifying how to coordinate its behavior with the other agent) and independent degrees (signifying when coordination is beneficial) concurrently during the learning process, which means that the learning approach is fully on-line. This is significantly different from other existing approaches in which agents need a predefined period to learn the situations when coordination is beneficial so that CL can be carried out afterwards [20], [36].

4) As stated above, this paper restricts discussions to two-agent scenarios. Two solutions are possible to extend the approach to scenarios involving more than two agents. One is to differentiate the identity of each agent and let an agent estimate the independent degrees with each of other agents individually. In this case, a joint Q-value table is built for each pair of agents during the CL process. The other solution is simply to disregard agents’ identities and build a joint Q-value tables for all the agents. Only an overall independent degrees is estimated during learning in this case. This solution is based on the idea that when an agent detects a conflict, this agent might not care who caused this conflict. The former solution seems to have the potential to achieve a better performance than the latter but costs a higher computational complexity.

V. EXPERIMENTAL STUDIES

Experimental studies have been carried out to test the effectiveness of the proposed approach, denoted as independent degrees learning (IDL), to solve robot navigation problems. The aim of the experiment is to test whether an efficient coordinated policy can be learnt through dynamic adaptation of agent independence without the assumptions of agents’ prior knowledge about the domain structure or the agents’ full observability of the environment.

A. Experimental Setting

1) Benchmark Approaches: Because most existing approaches to coordinated MAL in loosely coupled MASs are based on certain preconditions (e.g., robots’ global observability or prior knowledge about the domain structure), a direct comparison of the proposed approach with these approaches is not applicable. Three other approaches are selected as benchmarks for comparison of learning performance.

a) IL: In this approach, each agent treats the remaining agents simply as part of the environment, and learns its policy independently. The decision-making process is forcibly decomposed into n separate MDPs. This results in a large reduction in the state-action representation. However, at the same time, a poor learning outcome might occur because of the lack of coordination. IL provides a valid perspective to study the so-called “moving target” effect complicating general MAL problems. Despite the lack of guaranteed optimal performance, this approach has been applied successfully in certain cases [41], [42].

b) Joint state action learning (JSAL): This approach provides another extreme scenario opposed to IL. Agents either communicate freely with a central controller and select their individual actions according to those indicated by the central controller, or have full observability of the environment to receive the joint-state-action information of all agents to control the learning process synchronously. With sufficient learning periods, JSAL are capable of achieving an optimal performance as the overall decision-making process is considered as a single MDP, in which agents learn in a static environment. JSAL faces some tricky issues in MAL, including (1) the curse of dimensionality: the search space grows rapidly with the complexity of agent behaviors, the number of agents involved and the size of domains; (2) limited observability and restricted communication capability: agents might not have access to the needed information for the learning update because they are not able to observe the states, actions and rewards of all other agents; and (3) slow convergence: it may take many steps to explore all joint actions for every state, resulting in a slow convergence to the optimal policy.

c) CL [36], [43]: Approach CL is a state-of-the-art learning approach proposed to solve coordinated MAL problems in loosely coupled MASs, without assumptions of agents’ prior knowledge about the domain structure or the agents’ full observability of the environment. This approach first adopts a statistical method to detect coordinated states where coordination is most necessary. Then, according to the different transitions between independent and coordinated states, agents update Q values based on their limited observability of the environment by giving an optimistic estimation of the unobserved agents. CL is an off-line learning approach in terms that agents learn where and how to coordinate their behaviors separately. A state in approach CL is considered to be either an independent or a coordinated state. Comparing with this approach can thus give us a better understanding of the benefit by considering the uncertainties of agent independence using approach IDL.

2) Parameter Settings: In this experiment, the four approaches are first tested in three small domains, as shown in Fig. 2. In these domains, the state space is relatively small and in some regions of the domains, independent degrees might be quite low, indicating that coordination is heavily needed in order to avoid conflicts. These approaches are then applied to some larger domains, as shown in Fig. 4 [34]. There are fewer
interactions between robots in these large domains than in the small domains in Fig. 2.

When robots collide with a wall, they rebound and stay where they were. If they collide with each other, both robots break down and are transferred back to their original states. The exploration policy adopted is the fixed $\varepsilon$-greedy policy with $\varepsilon = 0.1$ to indicate a small probability of exploration. Learning rate $\alpha = 0.05$, discount factor $\gamma = 0.95$, trace-decay parameter $\lambda = 0.8$, perception distance $R = 2$, and rewards $r$ are given as follows: +20 for reaching the goal state, −1 for colliding with a wall and −10 for colliding with the other robot. All approaches are run for $N = 10,000$ episodes and average the last $N_{\text{avg}} = 2000$ episodes to compute the overall performance. To use approach CL, $\delta$ is set to be 95%. Two cases of CL are studied. In the first case of CL, which is denoted as $\text{CL}_1$, the scanning distance $R_1$ is set to 2, while in the second case, which is denoted as $\text{CL}_2$, the scanning distance $R_2$ is set to 4. In both cases, the first 1000 (i.e., $N_{\text{off}} = 1000$) episodes are used to collect statistical information to determine the coordinated states in approach CL. All results are then averaged over $\text{Run} = 25$ runs. The values and meanings of parameter settings are listed in Table I for clarity.

**B. Results and Analysis**

Fig. 5(a)–(c) shows the learning dynamics of the four learning approaches in small domains. We can see that approach JSAL can obtain an optimal reward because robots learn their policies based on joint state-action information. The IL approach, instead, can only achieve a very low reward due to the lack of coordination. That is, when a robot is learning independently, the robot cannot take into consideration the nonstationary of the learning environment. Thus, robots must continuously “catch up” with the dynamic and adaptive environment, causing a sub-optimal performance much lower than that in approach JSAL. Robots using approach CL conduct their learning independently during the first 1000 episodes and collect statistical information to determine the coordinated states where coordination is most required. After that, the average reward dramatically increases as robots can use the joint state-action information to lower the probability of conflicts when they are in the coordinated states. The performance in approach CL, however, still remains at a sub-optimal level due to an over estimation caused by the optimistic estimation mechanism (for more detail, see [36]), which always chooses the highest Q value based on the available information. Robots using the proposed approach IDL also conduct their learning independently at the beginning of the learning process because no dependent relationships between robots are assumed beforehand. The robots then adapt the dependent interactions between robots.
relationships (through adapting independent degrees) dynamically during the learning process. In this way, coordination can be achieved when there are low independent degrees in some states indicating that robots are mutually dependent on each other for decision-making. As can be seen in Fig. 5(a)–(c), the learning curves of approach IDL converge more quickly than those of approach JSAL at the beginning of learning process, and then gradually converge to the optimal values of approach JSAL. Robots using approach IDL perform better than robots using approach CL, even in the latter, robots have a longer scanning distance of $R = 4$ and learn based on joint state-action information in the coordinated states.

The overall results in small domains are given by Table II. To show the different computational complexity of these approaches, the state/action size and corresponding number of estimated Q values are also laid out. As can be seen from Table II, approach IDL reduces the computational complexity slightly compared with approach CL and significantly compared with approach JSAL. The significant reduction from approach JSAL is more desirable in large-scale domains where the computational complexity of JSAL is too high to be implemented, which can be verified by the results in large domains given later. In all small domains, the proposed approach IDL can achieve a reward that is much higher than the reward in approach IL and is quite near to the optimal reward in approach JSAL. This is because that approach IDL can capture the different levels of independence between robots for an efficient learning so that coordination can be achieved to avoid conflicts in certain parts of the environment. The results of collision percentage show that approach IDL decreases the likelihood of collision compared with approaches CL and IL. In approach JSAL, the optimal reward is a bit lower than 20 and robots may also have a low probability of collision. This is because of the stochastic exploration during the last 2000 episodes, when average performance is calculated. The steps of both robots to reach their own goals are calculated for those episodes when both robots do not collide with each other and reach the goal successfully. The results show that robots in approach IL always find the shortest path to their goals, which, in turn, causes the high probability of collision because they do not coordinate with each other when both come to potentially dangerous states with low independent degrees. On the contrary, approach JSAL receives the joint state-action information of both robots so that a safe detour strategy will be learnt by both robots to reduce the probability of collision, thus increasing the steps to the goals. The 95% confidence intervals of step number to goals also imply that approach JSAL is the most unstable approach, that is, the policy in approach JSAL can be affected by the stochastic learning process. This means if robots are “lucky” enough at the early exploitation stage, they can learn a short path to the goals. On the contrary, a much longer path is learnt if the learning process is deviated from a better trajectory. Approach IDL, however, combines the merits of both approach IL and JSAL, allowing robots to find the shortest path to the goals with a higher certainty while only making small detours around the states where coordination is most needed. This is why the step number to goals in approach IDL is a bit higher than that in approach IL but much lower than that in approach JSAL.

Fig. 6 gives a vivid illustration of robots’ transitions (not considering the collision with wall caused by the exploration during the learning process) in the HG domain to give a better understanding of the different learning performance of all the approaches. During the last 2000 episodes in each run, 50 episodes were randomly sampled and the robots’ transitions were traced to see how the robots managed to solve the collision conflict near the doorway in the HG domain. As expected, as robots using approach IL learn their policies independently, they cannot always distinguish with certainty whether it is safe to go through the doorway. As a result, both robots choose to pass the doorway simultaneously, causing the high percentage of collisions as illustrated by Fig. 6(a). Robots using approach CL $(R = 2)$ can determine coordinated states (i.e., the three states near the doorway) off-line and combine each other’s state and action information for coordinated learning in these states. That is why robot $R_2$ can learn to make a tour by transiting to a point beyond the doorway [i.e., transition 1 in Fig. 6(b)]. However, because the new state of robot $R_2$ is not in the learnt coordinated states, robot $R_2$ cannot receive robot $R_1$ ’s information for coordination in this new state. Although most of the time, robot $R_2$ can learn to avoid conflict with robot $R_1$ by transiting to the right side, in some cases, robot $R_2$ will still transit back to the coordinated states [i.e., transition 2 in Fig. 6(b)], causing a collision near the doorway. This probability of collision, however, is greatly reduced compared to that in approach IL. Fig. 6(c) shows the most likely occurring transitions when robots use approach IDL. As robots can learn a value of independent degree in each state, they can combine each other’s state information for coordination during each transition (with a different probability indicated by the value of the corresponding independent degree). As can be seen from Fig. 6(c), robot $R_2$ can

\begin{table}[h]
\centering
\caption{Performance of Different Learning Approaches in Small-Scale Domains (With 95% Confidence Intervals)}
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
\textbf{Domains} & \textbf{Approach} & \textbf{States} & \textbf{Actions} & \textbf{Q Values} & \textbf{Reward} & \textbf{Std. Error} & \textbf{Step} & \textbf{Conf. Int.} \\
\hline
\hline
\textbf{IL} & 21 & 4 & 84 & 0.10 (0.29) & 0.67 (0.02) & 25.18 (0.72) \\
\textbf{CL} & 22.9 & 7.7 & 128.48 & 18.28 (0.35) & 0.14 (0.02) & 27.54 (0.56) \\
\textbf{IDL} & 25.3 & 7.6 & 135.2 & 18.75 (0.26) & 0.13 (0.02) & 28.93 (0.35) \\
\textbf{JSAL} & 25.4 & 7.4 & 136.4 & 18.19 (0.39) & 0.49 (0.09) & 28.73 (0.59) \\
\hline
\textbf{HG} & 464 & 16 & 7.95 & 155.88 (0.49) & 0.07 (0.02) & 15.34 (2.27) \\
\hline
\textbf{TTG} & 25.3 & 17.7 & 164.16 & 18.63 (0.08) & 0.16 (0.08) & 14.82 (0.72) \\
\textbf{IDL} & 42.5 & 7.6 & 172.8 & 18.98 (0.16) & 0.09 (0.01) & 20.73 (0.09) \\
\hline
\textbf{JSAL} & 28.27 & 4 & 117.08 & 17.76 (0.10) & 0.06 (0.01) & 17.79 (0.22) \\
\hline
\textbf{TA} & 23.6 & 16 & 104.38 & 18.25 (0.19) & 0.09 (0.01) & 15.34 (0.93) \\
\hline
\end{tabular}
\end{table}
Fig. 7 shows the estimated values of $1 - \xi$ in three small domains from the perspective of robot $R_1$. (a) HG. (b) TTG. (c) TR.

Table III gives the overall performance of these four learning approaches in large-scale domains (with 95% confidence intervals).

In summary, the proposed approach IDL outperforms the uncoordinated approach IL by considering coordination only when the learnt robot independence indicates that this coordination is necessary. On the other hand, approach IDL reduces the computational complexity considerably and enables robots to learn a shorter path to the goal with a higher certainty than approach JSAL. Furthermore, approach IDL performs better than the state-of-the-art approach CL by considering the uncertainties of robot independence and without acquiring joint-action information in coordinated learning. Experimental results showed that approach IDL could efficiently learn the independent degrees in different states in order to quantify agent independence so that an efficient coordination policy could be devised with low computational complexity.

FIG. 7 Values of $1 - \xi$ in small-scale domains (robot $R_1$). (a) HG. (b) TTG. (c) TR.
VI. RELATED WORK

Much attention has been paid to the problem of coordinated MAL in recent multiagent research. Numerous approaches have been proposed to exploit the structural or networked dependence of agents for efficient coordinated learning. Hierarchical MAL approaches \[16\], \[44\] have been proposed to solve agent coordination problem where an overall task can be subdivided into a hierarchy of subtasks, each of which is restricted to the states and actions relevant to some particular agents. Coordination-graph based approaches \[19\], \[24\] took advantage of an additive decomposition of the joint reward function to local value functions, each of which is related only to a small number of neighboring agents, to efficiently construct jointly optimal policies. Distributed value functions based learning approach \[21\] exploited the dependence between networked agents so that each agent can learn a value function to estimate a weighted sum of future rewards of all the agents in a network. All the above approaches, however, focused on solving coordinated MAL when an explicit representation of agent independence was given beforehand. The agent independence is represented either through a predefined decomposition of the main task or a fixed interaction network topology. This paper differs from these approaches in that no explicit representation of the agent independence is assumed so that agents should build such a representation from experience.

Many approaches have been proposed with an assumption of agents’ full observability of the domain state for efficient coordinated learning. An approach called sparse tabular Q-learning \[45\] was proposed to learn joint action values on those coordinated states where coordination is beneficial. The action space is reduced significantly because agents can learn individually without taking into account the other agents in most situations, but only need to conduct coordinated learning in the joint action space when dependence between agents exists. These coordinated states where agents are mutually-dependent, however, need to be specified beforehand and the agents are assumed to have prior knowledge of these states. Kok et al. \[20\] and Kok and Vlassis \[24\] further extended their approach to enable agents to coordinate their actions when there are more complicated dependencies between agents, and used statistical information about the obtained rewards to learn these dependencies. All these studies are based on the framework of collaborative multiagent MDP \[46\], in which agents make decisions by using the joint state information. This full observability of world state for each individual agent is not assumed in this paper as the problems studied here are framed as Dec-MDPs, where each agent has only local observability so that the joint state cannot always be determined through an agent’s local information.

Some approaches have been developed in recent years with the aim of learning when coordination is beneficial in a loosely coupled MAS (particularly in robot navigation problems). De Hauwere et al. \[22\] proposed a solution to coordinated learning problems called 2observe. This approach decouples a MAL process into two separate layers. One layer learns where it is necessary to observe the other agent and the other layer adopts a corresponding technique to avoid conflicts. This approach, however, assumed that agents have full observability of the environment, which is undesirable in many systems. In our approach, agents only have limited observability of the environment and a joint-state learning process is conducted during the coordinated learning process. Spanel and Melo \[35\] introduced a model for solving the coordination problem in loosely coupled MASs called interaction-driven Markov games (IDMG). In IDMG, the states where agents should coordinate with each other are specified in advance and a fully cooperative Markov game is defined in these coordinated states such that agents can compute the game structure and Nash equilibria to choose their actions accordingly. IDMG is based on the game-theory solution to resolve the planning problem that requires the computation of multiple equilibria, which is computationally demanding. Later, Melo and Veloso \[47\] proposed a two-layer extension of the Q-learning algorithm to enable agents to learn where coordination is beneficial by augmenting the action space with a pseudo-coordination action. In their approach, agents are able to learn a trade-off between the benefits arising from good coordination and the cost of choosing the pseudo-coordination action. As a result, agents can learn to use the pseudo-coordination action in states only when it is necessary. The learning performance in \[47\], however, can be affected by the cost of choosing the pseudo-coordination action. In our approach, the coordination action is activated by the independent degrees, which carry the historical information indicating that such a coordination action will potentially bring more benefits. Our approach is fully event-driven, and the learning performance is not affected by the coordination action. De Hauwere et al. \[23\] proposed an algorithm called CQ-learning to enable agents to adapt the state representation in order to coordinate with each other. CQ-learning, however, depends on the assumption that each agent has already had an optimal individual policy so that every agent can have a model of its expected rewards. In this way, these states where the expected rewards differed significantly from the observed rewards are marked as dangerous states in which the other agent’s state information should be considered for decision-making. The assumption of individually learnt optimal policy is not required in our approach. Recently, an off-line learning approach was proposed to solve the robot navigation problems \[36\], \[43\]. This off-line learning first adopts a statistical method to detect those states where coordination is most necessary. An optimistic estimation mechanism is then applied to coordinate agents’ behaviors based on their limited observability of the environment. The approach proposed in this paper, instead, considers the uncertainties of agent independence and enables agents to learn the necessity of coordination and coordinated behaviors concurrently. Moreover, our approach does not use any joint-action information in the coordinated learning process.

Achieving coordination in MASs has also been extensively studied from the aspects of decentralized adaptive control systems. For example, a decentralized control strategy has been proposed in \[48\] that enables a group of robots to dynamically adapt their coverage so as to spread out over...
an environment while aggregating in areas of high sensory interests. A fully adaptive decentralized controller of robot manipulators for trajectory tracking was proposed in [49]. Many other studies (e.g., [50]–[52]) focused on decentralized adaptive control of synchronization (i.e., coordination) in complex MASs with stochastic dynamics or interconnection uncertainties. This paper differs from all these work because the decentralized coordination problem in this paper is modeled as Dec-MDPs, which is against the work in decentralized adaptive control systems in which agents’ state equation is represented in the form of a differential/difference equation. In [53], a multi-agent reinforcement learning approach was proposed to facilitate the coordinated learning process in grid-world domains when value functions are not shared among agents. This paper, however, did not address learning where to coordinate by exploiting agent independence.

VII. CONCLUSION

This paper proposed an MAL approach to solve coordination problems by exploiting agent independence in loosely coupled MASs. The approach enables agents to learn an efficient coordinated policy through dynamic adaptation of the estimation of agent independence. The proposed approach requires neither prior knowledge about the structure of the domain (e.g., the location of coordination) nor assumptions about the learning agents (e.g., full observability or optimal individual policies). These features set our approach apart from most existing approaches in related work and render it potentially suitable for wider practical applications. Experimental results showed that the proposed approach could guarantee a near-optimal performance in different scales of domains with low computational complexity.

In this paper, independent degrees are adjusted immediately when a conflict occurs. This mechanism can be improved by using a delayed updating rule to adjust the independent degrees based on a specified period of learning experiences. This issue is left for future research.

REFERENCES


Minjie Zhang (SM’13) received the B.Sc. and M.Sc. degrees from Fudan University, Shanghai, China, and the Ph.D. degree in computer science from the University of New England, Armidale, NSW, Australia.

She is currently a Professor of Computer Science with the University of Wollongong, Wollongong, NSW, Australia. She is an Active Researcher and has published over 100 papers in the past ten years. Her current research interests include multiagent systems and agent-based modeling in complex domains.

Fengbui Ren received the B.Sc. degree from Xidian University, Xi’an, China, in 2003, and the M.Sc. and Ph.D. degrees from the University of Wollongong, Wollongong, NSW, Australia, in 2006 and 2010, respectively.

He is currently an Australia Research Council Discovery Early Career Researcher Award in Australia Fellow and a Lecturer with the School of Computer Science and Software Engineering, University of Wollongong. He is an Active Researcher and has published over 50 research papers. His current research interests include agent-based concept modeling of complex systems, data mining and pattern discovery in complex domains, agent-based learning, smart grid systems, and self-organizations in distributed and complex systems.

Guozhen Tan received the B.S. degree from the Shenyang University of Technology, Shenyang, China, the M.S. degree from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree from the Dalian University of Technology, Dalian, China.

He is currently a Professor and the Dean of the School of Computer Science and Technology, Dalian University of Technology. He is also the Director of the Engineering and Technology Research Center for the Internet of things and Collaborative Sensing.

Liao Ning province. He was a Visiting Scholar at the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA, from 2007 to 2008. His current research interests include the Internet of things, cyber-physical systems vehicular Ad-hoc networks, intelligent transportation systems, and network optimization algorithms.

Prof. Tan was the recipient of the 2006 National Science and Technology Progress Award (second class) for his work in vehicle position and navigation, location-based service, traffic signal control, rapid responsiveness, and processing for traffic emergency. He was an Editor for the *Journal of Chinese Computer Systems*. He was a member of the China Computer Federation (CCF) and a committee man of the Internet Professional Committee of CCF, the Professional Committee of Software Engineering of CCF, and the Professional Committee of High Performance Computing of CCF.