

## Research Article

# A Decentralized Partially Observable Markov Decision Model with Action Duration for Goal Recognition in Real Time Strategy Games

Peng Jiao, Kai Xu, Shiguang Yue, Xiangyu Wei, and Lin Sun

*College of Information System and Management, National University of Defense Technology, Changsha 410073, China*

Correspondence should be addressed to Kai Xu; [xukai09@nudt.edu.cn](mailto:xukai09@nudt.edu.cn)

Received 22 March 2017; Accepted 8 June 2017; Published 16 July 2017

Academic Editor: Filippo Cacace

Copyright © 2017 Peng Jiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multiagent goal recognition is a tough yet important problem in many real time strategy games or simulation systems. Traditional modeling methods either are in great demand of detailed agents' domain knowledge and training dataset for policy estimation or lack clear definition of action duration. To solve the above problems, we propose a novel Dec-POMDM-T model, combining the classic Dec-POMDP, an observation model for recognizer, joint goal with its termination indicator, and time duration variables for actions with action termination variables. In this paper, a model-free algorithm named cooperative colearning based on Sarsa is used. Considering that Dec-POMDM-T usually encounters multiagent goal recognition problems with different sorts of noises, partially missing data, and unknown action durations, the paper exploits the SIS PF with resampling for inference under the dynamic Bayesian network structure of Dec-POMDM-T. In experiments, a modified predator-prey scenario is adopted to study multiagent joint goal recognition problem, which is the recognition of the joint target shared among cooperative predators. Experiment results show that (a) Dec-POMDM-T works effectively in multiagent goal recognition and adapts well to dynamic changing goals within agent group; (b) Dec-POMDM-T outperforms traditional Dec-MDP-based methods in terms of precision, recall, and  $F$ -measure.

## 1. Introduction

Recently, more and more commercial real time strategy (RTS) games have received attention from AI researchers, behavior scientists, policy evaluators, and staff training groups [1]. A key aspect in developing these RTS games is to create human-like players or agents who can act or react intelligently against changing virtual environment and incoming interactions from real players [2]. Though many AI planning and decision-making algorithms have been applied to agents in RTS games, their behavior patterns are still easy to be predicted and thus making games less entertaining or intuitive. This is partially because of agents' low information processing and understanding ability, for example, the recognition of goal or intention from opponents or friends. In other words, understanding goals or intentions in time helps agents cooperate better or make counter decisions more efficiently.

A typical scenario in RTS games is a group of AI players cooperating to achieve a certain mission. In the Star-Craft,

for example, the AI players have to cooperate so as to besiege enemy bases or intercept certain logistic forces [3]. Therefore, if AI players can recognize the real moving or attacking target, they will be better prepared, no matter with early defense employment or counter decision-making. Considering these benefits, goal recognition has attracted lots of attention from researchers in many different fields. Many related models and algorithms have been proposed and applied, such as hidden Markov models (HMMs) [4], conditional random fields (CRFs) [5], Markov decision processes (MDPs) [6], and particle filtering (PF) [7].

Hidden Markov models [8] are especially known for their applications in temporal pattern recognition such as speech, handwriting, and gesture recognition. Though convenient in representing system states, HMMs have low ability in describing agent actions in dynamic environment. Comparing to HMMs, MDPs have a better representation of actions and their future effects. MDP is the framework for solving sequential decision problems: agents select actions

sequentially based on states and each action will have an impact on future states. They have been successfully applied in goal and intention recognition [6]. Several modifications based on the MDP framework have a finer formalization towards more complex scenarios. Among these models, the Dec-POMDM (decentralized partially observable Markov decision model) [9] is a MDP-based method focusing on solving multiagent goal recognition problem. Though having all details of cooperation embedded in team's joint policy, Dec-POMDM is only concerned about actions starting and terminating within one time step. This is usually not applicable in RTS games.

Based on ideas from Dec-POMDM and SMDPs [10], we propose a novel decentralized partially observable Markov decision model with time duration (Dec-POMDM-T) to formalize multiagent cooperative behaviors with durative actions. The Dec-POMDM-T models the joint goal, the actions, and the world states hierarchically. Compared to works in [9, 11], Dec-POMDM-T explicitly models the time duration for primitive actions, indicating whether actions are terminated or not. In Dec-POMDM-T, the multiagent joint goal recognition consists of three components: (a) formalization of behaviors, the environment, and the observation for organizers; (b) model parameter estimation through learning or other methods; and (c) goal inference from observations:

- (a) For the problem formalization, agents' cooperative behaviors are modeled by joint policies, ensuring model's effectiveness without considering domain-related cooperation mechanism. Besides, explicit time duration modeling of primitive actions is also implemented.
- (b) For the parameter estimation, under the assumption of agents' rationality, many algorithms for Dec-POMDP could be exploited for accurate or approximate policy estimation, making the training dataset unnecessary. This paper uses a model-free algorithm named cooperative colearning based on Sarsa [12] in policy learning.
- (c) For the goal reference, the modified particle filtering method is exploited because of its advantages in solving goal recognition problems with different sorts of noises, partially missing data and unknown action duration.

Like the modified predator-prey problem presented in [9], the scenario in this paper also has more than one prey and predator. The predators first establish joint pursuing target or goal, which would be changed halfway, before capturing it. The model and its inference methods applied in this paper are to recognize the real goal behind agents' cooperative behaviors which are partially observable traces with additional noises. Based on this scenario, we retrieve agents' optimal policies using a model-free multiagent reinforcement learning (MAREL) algorithm. After that, we run a simulation model in which agents select actions according to policies and generate a dataset consisting of 100 labeled traces. With this dataset, statistical metrics including precision, recall, and F-measure are computed using Dec-POMDM-T and other

Dec-MDP-based methods, respectively. Experiments show that Dec-POMDM-T outperforms the others in all three metrics. Besides, recognition results of two traces are also analyzed, showing that Dec-POMDM-T is also quite robust when joint goals change dynamically during the recognition process. The paper also analyzes the estimation variance and time efficiency of our modified particle filter algorithm and thus proves its effectiveness in practice.

The rest of the paper is organized as follows. Section 2 introduces related works. Section 3 analyzes the moving process in RTS games and presents the formal definition of Dec-POMDM-T as well as its DBN structure. Based on that, Section 4 introduces the way to use modified particle filter algorithm in multiagent joint goal inference. After that, experiment scenarios and parameter settings as well as results are shown in Section 5. Finally, the paper draws conclusions and discusses future works in Section 6.

## 2. Related Works

As an interdisciplinary research hotspot covering psychology and artificial intelligence, the problem of goal recognition or intention recognition has been tried from many different ways. In early days, the formalization of goal recognition problem is usually related to the construction of plan library, in which the recognition process is based on logical consistency matching between observations and plan library. After that, the well-known Probabilistic Graphic Models (PGMs) [13] family, including MDPs [6], HMMs [3], and CRFs [5], were further proposed as a more compact graph-based representation approach. Additionally, PGMs have their advantage in modeling the uncertainty and dynamics both in environments and the agent itself, which is not possible in the above consistency-based methods. Among PGMs, several modifications including forming hierarchical graph model structure [14–16] and explicit modeling of action duration [17, 18] are also proposed. Although probabilistic methods have their advantage in uncertainty modeling, still they cannot represent and process structural or relational data. Statistical relational learning (SRL) [19] is a relatively new theory applied in intention recognition, including logical HMMs (LHMMs) [20], Markov logic networks (MLNs) [21], and Bayesian logic programs (BLPs) [22]. It combines relation representation, first-order logic, probabilistic inference, and machine learning altogether. Besides, several other methods based on probabilistic grammar have also been proposed on the discovery of the similarity between natural language process (NLP) and intention recognition [23]. Most recently, deep learning and other intelligent algorithms in retrieving agent's decision model are also applied in intention recognition [24]. Other considerations like goal recognition design (GRD) [25, 26] try to solve the same problem from different aspects.

*2.1. Goal Recognition with Action Duration Modeling.* There is a group of models in PGMs, like HMM-/MDP-based models, that has close relationship with Markov property. The property assumes that the future states depend only on the current state. Generally speaking, the Markov property enables reasoning and computation with the model that would otherwise

be intractable. Though it is desirable for models to exhibit Markov property, it is not always the truth in real goal recognition scenarios, causing serious performance degradation like lower precision, longer convergence time, and even wrong prediction. One main reason for Markov property violation occurs in agents having durative primitive actions. Typically there are two approaches in solving the above problem. One is forming hierarchical structures. Fine et al. [14] proposed Hierarchical HMM (HHMM) in 1998. Bui et al. [3] used abstract hidden Markov models (AHMM) for hierarchical goal recognition based on abstract Markov policies (AMPs). A problem of the AHMM is that it does not allow the top-level policy to be interrupted when the subplan is not completed. Saria and Mahadevan [27] extended the work by Bui to multiagent goal recognition. Similar modifications include works like Layered HMM (LHMM) [15], Dynamic CRF (DCRF) [28], and Hierarchical CRF (HCRF) [16].

Another kind of approaches tackling non-Markov property falls into explicit modeling of action duration time. Hladky and Bulitko [17] applied hidden semi-Markov model (HSMM) to opponent position estimation in the first person shooting (FPS) game Counter Strike. Duong et al. [18] proposed a Coxian hidden semi-Markov model (CxHSMM) for recognizing human activities of daily living (ADL). The CxHSMM modifies HMM in two aspects: on one hand, it is a special DBN representation of two-layer HMM, and it also has termination variables; on the other hand, it used Coxian distribution to model the duration of primitive actions explicitly. Besides, Yue et al. [9] proposed a SMDM (semi-Markov Decision Model) based on AHMM, which not only has hierarchical structure, but also models the time duration. Similar methods also include Semi-Markov CRF (SMCRF) [29] and Hierarchical Semi-Markov CRF (HSCRF) [30].

**2.2. Multiagent Goal Recognition Based on MDP Framework.** As what we have known, MDP is the framework for solving sequential decision problems. Baker et al. [6] proposed a computational framework based on Bayesian inverse planning for recognizing mental states such as goals. They assumed that the agent is rational: actions are selected based on an optimal or approximate optimal value function, given the beliefs about the world, and the posterior distribution of goals is computed by Bayesian inference. Ullman et al. [31] also successfully applied this theory in more complex social goals, such as helping and hindering, where an agent's goals depend on the goals of other agents. In the military domain, Riordan et al. [32] borrowed Baker's idea and applied Bayesian inverse planning to inferred intents in multi-Unmanned Aerial Systems (UASs). Ramirez and Geffner [11] extended Baker's work by applying the goal-POMDP in formalizing the problem. Compared to the MDP, the POMDP models the relation between real world state and observation of the agent explicitly. Comparing to POMDP, I-POMDP defines an interactive state space, which combines the traditional physical state space with explicit models of other agents sharing the environment in order to predict their behavior. Ramirez and Geffner also solved the inference problem even when observations are incomplete. Besides, Yue et al. [9] also proposed a Dec-POMDM model based on

Dec-POMDP in recognizing multiagent goal recognition. Its model, however, does not consider situations when agents are having durative actions in RTS games. Above modifications based on MDP framework, like SMDPs, POMDPs, and Dec-POMDPs, all have a finer formalization towards more complex scenarios.

### 3. The Model

We propose the Dec-POMDM-T for formalizing the world states, behaviors, goals, and action durations in goal recognition problem. In this section, we first introduce how agents do path planning and move between adjacent grids in RTS games. Then, the formal definition of the Dec-POMDM-T and relations among variables in the model is explained by a DBN representation. Based on that, the planning algorithm for finding out the optimal policies is given.

**3.1. Agent Maneuvering in RTS Games.** Agents' maneuvering in RTS games usually consists of two processes: one is the path planning knowing the starting point and destination beforehand; the other one is agents moving from current positions to adjacent grids.

**3.1.1. Path Planning.** Like many classical planning problems, path planning would also generate several courses of actions given starting points and destinations, which is a sequence of positions specifically. In dynamic environments however, the effects of actions would be uncertain. Besides, agent maneuvering is essentially a sequential decision problem, in which agents select actions according to current states and destinations. Further, in multiagent cooperative behaviors, path planning also needs to follow joint policy shared among the agent group. Thus a probabilistic Markov decision model is needed.

**3.1.2. Moving between Adjacent Grids.** After knowing the next position or grid from path planning algorithm, agents need to move from original position to it. In real situations, one moving action usually lasts for several steps before agent arriving in target position. This situation breaks the Markov property, and thus making the agent decision process falls into a semi-Markov one.

As in Figure 1, which is originally shown in [33], assume that an agent is on the point X which is in the grid of C2 and wants to go to the point Z which is in the grid A3. For the path planning level, the agent needs to choose a grid among the five adjacent grids (B1, B2, B3, C1, and C3). In this example, the agent decides to go to grid B2. In the moving level, the agent will move along the line from point X to point S which is the center of grid B2. Because the simulation step is a short time, the agent will compute how long it will take to reach point S according to the current speed. Because the position of the agent is a continuous variable, it is very unlikely that the agent just gets the grid center when a simulation step ends. Thus, the duration of moving is usually computed by

$$\text{duration} = \left\lceil \frac{\|\text{position}_X - \text{position}_S\|}{\text{speed} \times T_{\text{step}}} \right\rceil, \quad (1)$$

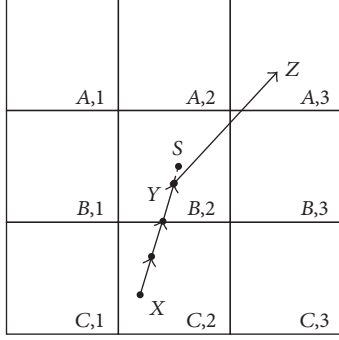


FIGURE 1: An example of maneuvering on a grid map.

where speed is a constant in the moving process and  $T_{\text{step}}$  is the time of a simulation step.  $\|\text{position}_X - \text{position}_S\|$  is the distance between the point  $S$  and point  $X$ . The duration is computed by a floor operator. In this case, duration = 3. After moving for 3 steps from position  $X$ , the agent will get position  $Y$  and choose the next grid. This moving process will not be intercepted except that the intention is changed.

**3.2. Formalization.** In standard definition of Dec-POMDP, there is no concept of intention or joint intention. The Dec-POMDP defines the states which consist of all information needed for making decisions. When formalizing a model for goal recognition, the original definition of states should be further decomposed into inner and external states, corresponding to agents' intentions and outside environment, respectively. Thus the action selection is determined by all inner and external states. Besides, in multiagent goal recognition for cooperative behaviors, inner states could further be extended to joint intentions or goals. In our Dec-POMDM-T, it should also satisfy situations when joint goal can be terminated as of goal achievement or halfway interruption. Thus, the Dec-POMDM-T is a combination of four parts: (a) the standard Dec-POMDP; (b) the joint goal and goal termination variable; (c) the observation model for recognizer; and (d) the time duration for joint actions and action termination variables.

A classic Dec-POMDP is a tuple  $\langle I, S, A, \Omega, \vec{a}, \vec{o}, \vec{\pi}, P_T, P_O, \lambda, h, R \rangle$ , where

- (i)  $I$ : set of agents, indexed  $1, \dots, n$ ;
- (ii)  $S$ : set of states;
- (iii)  $A$ : set of joint actions,  $A = \otimes_{i \in I} A_i$ , in which  $A_i$  is the set of possible actions for agent  $i$ ;
- (iv)  $\Omega$ : set of joint observations,  $\Omega = \otimes_{i \in I} \Omega_i$ , in which  $\Omega_i$  is the set of observations for agent  $i$ ;
- (v)  $\vec{a}$ : joint action,  $\vec{a} = \langle a_1, \dots, a_n \rangle$ ;
- (vi)  $\vec{o}$ : joint observation,  $\vec{o} = \langle o_1, \dots, o_n \rangle$ ;
- (vii)  $\vec{\pi}$ : joint policy  $\vec{\pi} = \langle \pi_1, \dots, \pi_n \rangle$ , in which  $\pi_i$  is the local policy for agent  $i$ , mapping from  $\vec{o}^i = o_{i_1}, \dots, o_{i_t}$  to action  $a_i$ ;
- (viii)  $P_T$ : transition function,  $P_T = P(s' | s, a)$ ;

- (ix)  $P_O$ : observation function,  $P_O = P(o | a, s')$ ;
- (x)  $\lambda$ : discount factor;
- (xi)  $h$ : planning horizon;
- (xii)  $R$ : reward function,  $R = R(\vec{a}, s')$ .

More definition details, explanations, and demonstrations could be found in [34]. As we have discussed above, the original Dec-POMDP has no definition of joint goals, observation model for recognizers, and action durations.

Besides,  $\pi_i$  in Dec-POMDP, mapping from  $\vec{o}^i = o_{i_1}, \dots, o_{i_t}$  to action  $a_i$ , does not satisfy Markov property. Thus we simply assume that agents select actions based only on current states as work in [9, 11]. Therefore, the Dec-POMDM-T becomes a tuple  $\langle I, S, \vec{A}, \vec{\Omega}, \vec{a}, \vec{o}, \vec{\pi}', P_T, P_O, \lambda, h, R, G, E_1, g, Y, Z, C, B, \vec{D}, \vec{E}_2 \rangle$ , where

- (i)  $G$ : set of all possible joint goals,  $G = \otimes_{i \in I} G_i$ , in which  $G_i$  is the set of goals for agent  $i$ ;
- (ii)  $GE$ : the joint goal termination variable which is shared among agents in multiagent cooperative behaviors,  $E_1 = \{0, 1\}$ ;
- (iii)  $\vec{\pi}'$ : joint policy  $\vec{\pi}' = \langle \pi'_1, \dots, \pi'_n \rangle$ , in which  $\pi'_i$  is the local policy for agent  $i$ , mapping from  $\vec{o}^i = o_{i_t}$  to action  $a_i$ ;
- (iv)  $g$ : the joint goal shared among agents;
- (v)  $Y$ : the observation function for recognizer, which is defined as  $Y : S \times Y \rightarrow [0, 1]$ ;
- (vi)  $Y$ : the finite set of joint observations for recognizer;
- (vii)  $Z$ : the goal selection function, which is defined as  $Z : S \times G \rightarrow [0, 1]$ ;
- (viii)  $C$ : goal termination function, which is defined as  $C : S \times G \rightarrow [0, 1]$ ;
- (ix)  $B$ : the initial goal distribution at  $t = 0$ ;
- (x)  $\vec{D}$ : the set of time durations of actions,  $\vec{D} = \langle d_1, \dots, d_n \rangle$ , where  $d_i \in \{0, 1, 2, \dots\}$  and is a nature number indicating additional time steps needed for accomplishing agent  $i$ 's current action;
- (xi)  $\vec{AE}$ : set of action termination variables,  $\vec{AE} = \langle ae_1, \dots, ae_n \rangle$ , where  $ae_i$  tells whether the agent  $i$ 's current action is terminated.

In above definitions,  $E_1$  tells whether cooperative agents would pursue the current goal  $g$  in next time step or change it according to goal selection function  $Z$ .  $\vec{D}$  would be computed when new actions are taken according to (1) defined above. As discussed above,  $\vec{E}_2$  indicates the on and off situation of each action. It would be affected by both  $E_1$  and  $\vec{D}$ , which would be further explained in following sections.

**3.3. The DBN Structure.** Essentially, the Dec-POMDM-T is a dynamic Bayesian network, in which all causalities would be depicted. In this section, we first introduce some subnetworks



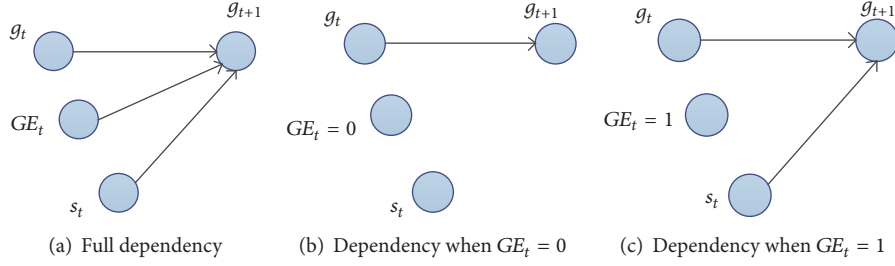


FIGURE 2: Subnetwork for joint goal.

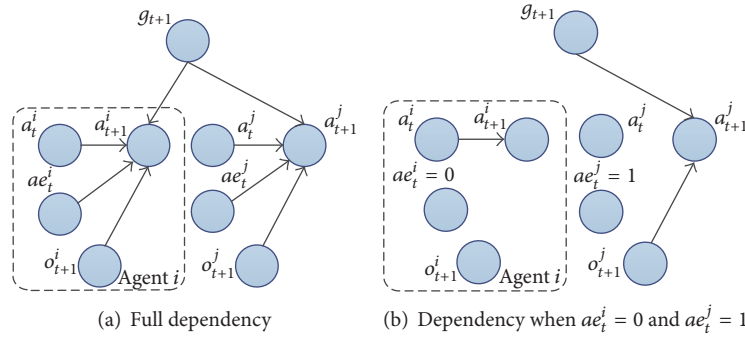


FIGURE 3: Subnetwork for action taking by different agents.

so as to explain their causal influences among different variables, like joint goal, states, actions, time durations, and termination variables. Based on that, a full DBN structure depicting two time slices of Dec-POMDM-T is presented.

Figure 2 shows the subnetwork for joint goal in cooperative missions. As shown in Figure 2(a), the full dependency of the joint goal  $g_{t+1}$  would include no more than original goal  $g_t$ , goal termination variable  $GE_t$ , and the current state  $s_t$  at time  $t$ . When  $GE_t$  takes on 0 at time  $t$ , showing that joint intention is not terminated,  $g_{t+1}$  would remain the same as  $g_t$ . While if  $GE_t$  takes on 1, agents would select another joint goal according to goal selection function  $Z$  with  $Z(s, \text{goal}) = p(\text{goal} | s)$ . In our modified predator-prey scenario, it means that predator team would change their joint target with the consideration of their inner and outer situations.

Similarly, we also depict the subnetwork for action taking by different agents in Figure 3. As shown in Figure 3(a), action selection for agent  $i$  at time  $t + 1$  would always be determined by the previous executing action  $a_t^i$ , action termination indicator  $ae_t^i$ , observation  $o_{t+1}^i$ , and the joint goal  $g_{t+1}$  at time  $t + 1$ . Different situations are described in Figure 3(b), with agent  $i$  continuing its action  $a_t^i$  when  $ae_t^i = 0$  and agent  $j$  taking new action  $a_{t+1}^j$  based on  $o_{t+1}^j$  and  $g_{t+1}$  when  $ae_t^j = 1$ . The action selection follows  $\pi_j^i$  for agent  $j$ .

Further, the relationships for action time duration are depicted in Figure 4. As it shows, the time duration  $d_{t+1}^i$  for action  $a_{t+1}^i$  would be determined by  $d_t^i$  according to  $p(d_{t+1}^i | d_t^i)$  if  $ae_t^i = 0$ . While when  $ae_t^i = 1$ , indicating  $a_t^i$  to

be terminated, a new  $d_{t+1}^i$  would be computed by  $p(d_{t+1}^i | a_{t+1}^i, o_{t+1}^i)$ .

Other variables and their parameters are given as follows.

- (i) *Goal Termination Variable* ( $GE_t$ ).  $GE_t$  depends on  $g_t$  and  $s_t$  with  $p(GE_t | g_t, s_t)$ .
- (ii) *Action Termination Variable* ( $ae_t^i$ ).  $ae_t^i$  depends on  $GE_t$  and  $d_t^i$  with  $p(ae_t^i | GE_t, d_t^i)$ .
- (iii) *Agent Observations* ( $o_t^i$ ).  $o_t^i$  is the reflection of real state  $s_{t-1}$  with  $p(o_t^i | s_{t-1})$ .
- (iv) *Recognizer Observations* ( $y_t$ ).  $y_t$  is the observation of recognizers with  $p(y_t | s_t)$ .

The full DBN structure of Dec-POMDM-T in two time slices is presented in Figure 5.

For simplicity and clarity, a snapshot of only the agent  $i$  in two time slices is presented in Figure 5, with its activities being depicted using dashed frame in both slices. Detailed relationships among variables have already been explained in Figure 2 to Figure 5. Agents have no knowledge about each other and make their decision based on individual observations. Apparently, the DBN structure of the Dec-POMDM-T is much more complex than previous works in [3, 9, 33]. Compared to goal or plan recognition models with hierarchical structures like AHMM [3] and SMDM [33], the Dec-POMDM-T implicitly represents task decomposition

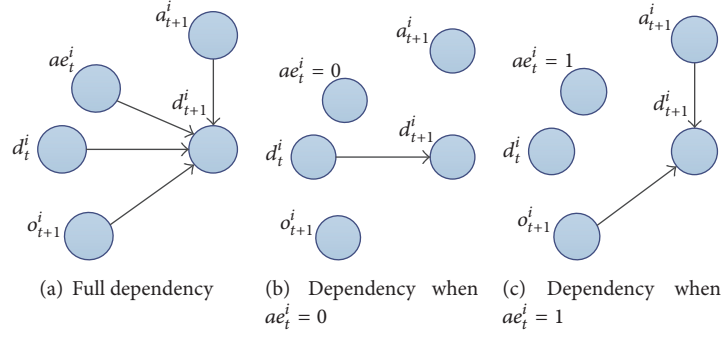


FIGURE 4: Subnetwork for action time duration.

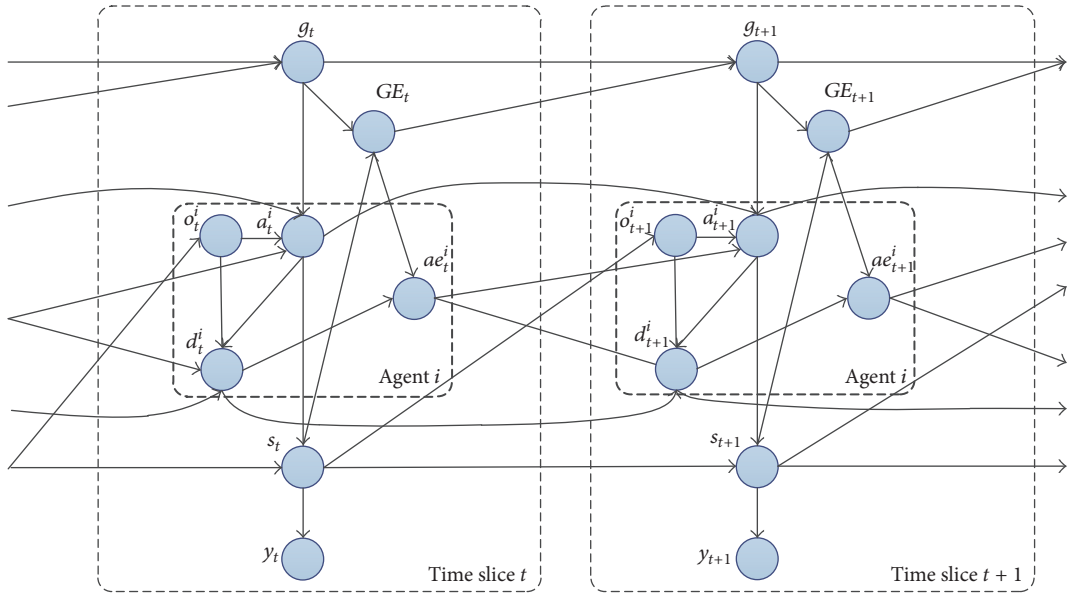


FIGURE 5: The DBN structure of the Dec-POMDM-T.

and mission allocation in joint policies. While for models [9] based on Dec-POMDP, the Dec-POMDM-T explicitly models time duration of primitive actions.

#### 4. Inference

Recognizing the multiagent joint goal is an inference problem trying to find out the real joint goal behind agent actions based on observations online. Essentially, this process is to compute the distribution of joint goal  $g_t$  given  $y_t$ , which is  $p(g_t | y_t)$ . It can be achieved either by accurate inference methods or approximate ones. As we have already exhibited the complexity of Dec-POMDM-T's DBN structure in above section, accurate inference of  $p(g_t | y_t)$  would be quite time consuming and thus impractical in many RTS games. Besides, accurate inference requires nearly perfect observations which would also be impossible in RTS games permitting only partially observable data using similar applications of war fog.

Traditional methods like Kalman filter and HMM filter usually rely on various assumptions to ensure mathematical tractability. However, data in multiagent goal recognition

involves elements of non-Gaussianity, high-dimensionality, and nonlinearity and thus preclude analytic solutions. As a widely applied method in sequential state estimation, particle filter (PF) is a kind of sequential Bayesian filter based on Monte Carlo simulations [35]. Unlike methods like extended Kalman filter and grid-based filters, PF is very flexible, easy to implement, and applicable in very general settings. Besides, PF also has no restriction on types of system noises.

The working mechanism of classic particle filter is as follows. The state space is partitioned as many parts, with the particles being filled-in according to prior distribution of states. The higher the probability or weight is, the denser the particles are concentrated. All of particles evolve along the time according to state transitions, reflecting the evolution of state estimation. The weights of particles would then be updated and normalized. Further, particles are resampled after a certain period as a countermeasure for sample impoverishment. The above description is a standard SIS (Sequential Importance Sampling) particle filter with resampling, consisting of four steps, including initialization, importance sampling, weight update, and particle resampling. The essence of PF is to empirically represent a posterior

distribution or density using a weighted sum of  $N_p$  samples drawn from the posterior distribution

$$p(x_t | y_t) \approx \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(x_t - x_t^{(i)}) \equiv \hat{p}(x_t | y_t), \quad (2)$$

where  $x_t^{(i)}$  are assumed to be *i.i.d* drawn from  $p(x_t | y_t)$ . When  $N_p$  is large enough,  $\hat{p}(x_t | y_t)$  approximates the true posterior distribution  $p(x_t | y_t)$ . The importance weights  $W_n^{(i)}$  can be updated recursively:

$$\begin{aligned} W_n^{(i)} &= \frac{p(x_{0:t}^{(i)} | y_{0:t})}{q(x_{0:t}^{(i)} | y_{0:t})} \\ &\propto W_{n-1}^{(i)} \frac{p(y_t | x_{t-1}^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{0:t})}. \end{aligned} \quad (3)$$

When the PF is applied in multiagent goal recognition under the framework of Dec-POMDM-T, the set of particles is defined as  $\{x_t^{(i)}\}_{i=1:N_p}$ , where  $x_t^{(i)} = \langle g_t^{(i)}, GE_t^{(i)}, o_t^{(i)}, \vec{a}_t^{(i)}, \vec{d}_t^{(i)}, \vec{ae}_t^{(i)} \rangle$ .  $N_p$  is the number of particles and the weight of  $i$ th particle is  $W_t^{(i)}$ . As we use the simplest sampling, the  $q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{0:t})$  is set to be  $p(x_t^{(i)} | x_{t-1}^{(i)})$ . And as the observation  $y_t$  only depends on  $s_t$ , the importance weight  $W_t^{(i)}$  can be updated by

$$W_t^{(i)} = W_{t-1}^{(i)} \cdot p(y_t | s_t). \quad (4)$$

The detailed procedure of multiagent goal recognition under the framework of the Dec-POMDM-T is given in Algorithm 1.

Four classic components of the SIS PF with resampling are all present in Algorithm 1, with particle initialization from line (2) to line (4), sequential importance sampling from line (6) to (25), weight updating and normalizing from line (26) to (29), and particle resampling in line (30). The joint goal sampling in line (10) follows  $g_t^{(i)} \cdot p(g_t^{(i)} | g_{t-1}^{(i)}, s_t^{(i)})$ . The observation for agents follows  $\vec{o}_t^{(i)} \cdot p(\vec{o}_t^{(i)} | s_{t-1}^{(i)})$  as in line (12). The joint goal termination samples are from  $GE_t^{(i)} \cdot p(g_t^{(i)} | s_t^{(i)})$  in line (13). Time duration for action  $a_t^{(i)(j)}$  would be updated following  $d_t^{(i)(j)} \cdot p(d_t^{(i)(j)} | d_{t-1}^{(i)(j)})$  in line (17).

Also, action changes would be sampled from  $a_t^{(i)(j)} \cdot p(a_t^{(i)(j)} | g_t^{(i)}, o_t^{(i)(j)})$  in line (19). Compute the action time duration of  $a_t^{(i)(j)}$  following  $d_t^{(i)(j)} \cdot p(d_t^{(i)(j)} | a_t^{(i)(j)}, o_t^{(i)(j)})$  as in line (20). Further, sample the action termination following  $ae_t^{(i)(j)} \cdot p(ae_t^{(i)(j)} | GE_t^{(i)}, d_t^{(i)(j)})$  in line (22). Each agent performs its action and changes the states accordingly. In the resampling process, the algorithm first calculates  $\hat{N}_{\text{eff}}$  according to

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_p} (\hat{W}_t^{(i)})^2}. \quad (5)$$

The resampling process returns if  $\hat{N}_{\text{eff}} > N_T$ , where  $N_T$  is the predefined threshold which could be  $N_p/3$  or  $N_p/2$ ;

otherwise generate a new particle set  $\{x_t^{(i)}\}$  by resampling with replacement of  $N_p$  times from the previous set  $\{x_{t-1}^{(i)}\}$  with probabilities  $\hat{W}_t^{(i)}$ , and then reset the weights to  $1/N_p$ .

## 5. Experiments

**5.1. The Modified Predator-Prey Problem.** In this paper, a modified predator-prey problem [9] is used. Compared to the classic one, the modified one has more than one prey for more than one predator to catch. This gives the test bed for evaluating our multiagent goal recognition algorithm based on Dec-POMDM-T. Our aim is to recognize the real target of predators based on noisy observations.

Figure 6 shows the  $5\text{ m} \times 5\text{ m}$  map and the predator's observation model in modified predator-prey problem. There are two predators and two preys on the map, denoted by red triangle and blue diamond, respectively. Predators establish a joint goal by choosing one of the prey and work cooperatively to capture it. The predator's observation model has also been explained in Figure 6. As we know, agents using tactical sensors in RTS games usually have a noisy and partial observation. They know exactly what is happening around, but the information quality drops when the distance gets larger. This degeneration process is simply modeled by the red circle with its radius set to 2 m in Figure 6. Further, we use several vertical and horizontal lines to separate cardinal *directions* into N, NE, E, SE, S, SW, W, and NW, respectively. The directions inside the circle are denoted by "*direction\_1*," while those outside are denoted by "*direction\_2*." Thus the example in our  $5\text{ m} \times 5\text{ m}$  map is as follows. According to Predator A's observation, Prey B is close to it and locates in SE\_1 while Prey A and Predator B each locates in NW\_2 and S\_2. Predator B, however, has a clear sight of Prey B in the near northeast NE\_1, while Prey A and Predator A are all in a relatively far direction of NW\_2 and N\_2. All agents can move in four directions (north, east, south, and west) or stay at the current position. Rules are set to prevent agents from moving out of the map. The joint goal would be achieved when both of predators have less than 0.5-meter distance with their target. Predators' target, or joint goal, could be changed halfway. The observation model for the recognizer is that it can have exact positions of preys while getting noisy observation of predators. Our purpose is to compute the posterior distribution of predators' joint goal using observation traces.

Some important definitions in Dec-POMDM-T under this scenario are as follows.

- (i) *I*: the two predators;
- (ii) *S*: the positions of predators and preys;
- (iii) *A*: five actions for predators with moving in 4 directions and staying still;
- (iv) *G*: Prey A or Prey B;
- (v)  $\Omega$ : the directions of agents faraway and exact positions of agents nearby;
- (vi) *Y*: the real positions of prey and noisy positions of predators;
- (vii) *h*: planning horizon.

**Input:** particle number  $N_p$ , agent team size  $N_A$ , resampling threshold  $N_T$ .

```

(1) Set time steps  $t = 1$ .
(2) For  $i = 1, \dots, N_p$ 
(3)   sample  $x_t^{(i)} \sim p(x_t)$ , and set  $W_t^{(i)} = 1/N_p$ . % Initialization
(4) End For
(5) For  $t = 2, 3, \dots$ 
(6)   For  $i = 1, \dots, N_p$ 
(7)     If  $GE_{t-1}^{(i)} = 0$  % Check if joint goal terminate
(8)        $g_t^{(i)} \leftarrow g_{t-1}^{(i)}$ .
(9)     Else
(10)       $g_t^{(i)} \leftarrow \text{SampleJointGoal}(g_{t-1}^{(i)}, s_t^{(i)})$ .
(11)    End If
(12)     $\tilde{o}_t^{(i)} \leftarrow \text{Observe}(s_{t-1}^{(i)})$ .
(13)     $GE_t^{(i)} \leftarrow \text{SampleGoalTerminate}(s_t^{(i)})$ .
(14)    For  $j = 1, \dots, N_A$ 
(15)      If  $ae_{t-1}^{(i)(j)} = 0$ 
(16)         $a_t^{(i)(j)} \leftarrow a_{t-1}^{(i)(j)}$ .
(17)         $d_t^{(i)(j)} \leftarrow \text{TimeDurationUpdate}(d_{t-1}^{(i)(j)})$ .
(18)      Else
(19)         $a_t^{(i)(j)} \leftarrow \text{SampleActionChange}(g_t^{(i)}, \tilde{o}_t^{(i)(j)})$ .
(20)         $d_t^{(i)(j)} \leftarrow \text{ComputeTimeDuration}(a_t^{(i)(j)}, \tilde{o}_t^{(i)(j)})$ .
(21)      End If
(22)       $ae_t^{(i)(j)} \leftarrow \text{SampleActionTermination}(GE_t^{(i)}, d_t^{(i)(j)})$ .
(23)    End For
(24)     $s_t^{(i)} \leftarrow \text{Perform}(s_t^{(i)}, \tilde{a}_t^{(i)})$ . % Action Perform
(25)  End For
(26) For  $i = 1, \dots, N_p$ 
(27)   Calculate the importance weights  $W_t^{(i)} = W_{t-1}^{(i)} \cdot p(y_t | s_t)$ 
(28) End For
(29)  $W_t \leftarrow \text{Normalize}(W_t)$ . % Weight normalization
(30) Calculate  $\hat{N}_{\text{eff}}$ , return if  $\hat{N}_{\text{eff}} > N_T$ ; otherwise resampling
(31) End For

```

ALGORITHM 1: Multiagent joint goal inference based on SIS PF with resampling under Dec-POMDM-T.

**5.2. Experiment Settings.** In this section, we provide parameter settings in scenarios, policy learning, and goal inference algorithm.

**5.2.1. Scenario.** Preys have no decision-making ability. They are senseless and select all five actions randomly. The initial positions of agents are randomly generated. The initial goal distribution is set to be  $p(g_0 = \text{Prey A}) = 0.6$  and  $p(g_0 = \text{Prey B}) = 0.4$ . As the map is  $5\text{ m} \times 5\text{ m}$ , we set the moving speed to  $0.5\text{ m/step}$ .

The goal termination function is simplified in the following way. If predators capture their target, then the goal is achieved; otherwise the predator team would change their joint goal with a probability of 0.05 for every time step.

At each time step, the recognizer has half a chance of getting each predator's true position, with the other half chance being of getting noisy positions:

$$\text{NoisyPosition} = \text{TruePosition} + \text{Directions} \{i\} \times H, \quad (6)$$

where  $H$  and  $\text{Directions} = \{[1, 0], [1, -1], [0, -1], [-1, -1], [-1, 0], [-1, 1], [0, 1], [1, 1]\}$  each represents the vibration strength of observation noise and its 8 possible directions.

**5.2.2. Policy Learning.** Under the assumption of agents' rationality, the paper applies a model-free MARL algorithm, named cooperative colearning based on Sarsa, in learning agent's optimal policy. The core idea of the algorithm is to



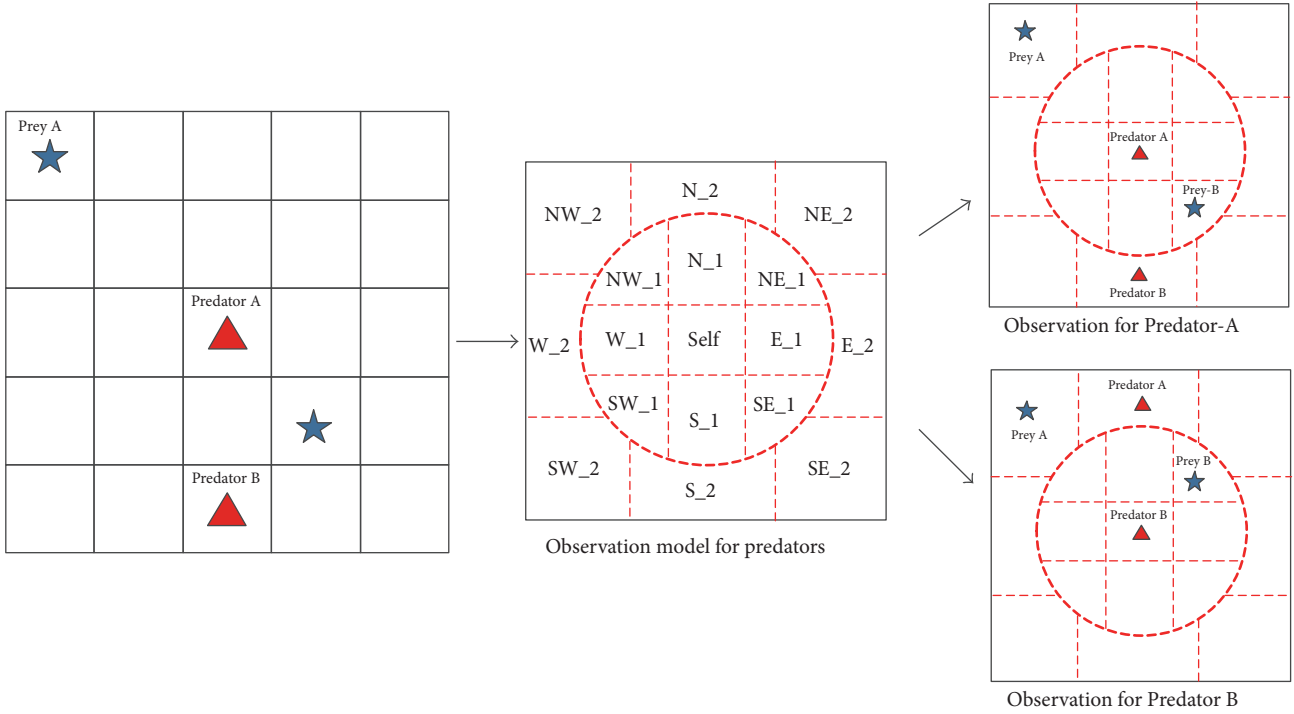


FIGURE 6: The 5 m × 5 m map and predators' observation model in predator-prey problem.

choose at each step a subgroup of agents and update their policies to optimize the task, given the fact that the rest of the agents have fixed plans; then, after a number of iterations, the joint policies can converge to Nash equilibrium.

The discount factor  $\lambda$  is set to 0.8. And the predator selects an action  $a_i$  given the observation  $o_t^i$  with a probability

$$p(a_i | o_t^i) = \frac{\exp(Q(o_t^i, a_i) / \beta)}{\sum_{i=1}^5 \exp(Q(o_t^i, a_i) / \beta)}, \quad (7)$$

where  $\beta = 0.1$  is the Boltzmann temperature. We set  $\beta > 0$  as a constant, which means that predators would always select approximately optimal actions. In our scenarios, the  $Q$ -value would converge after 750 iterations. In the learning process, if predators cannot achieve their goal in 5000 steps, the process would be reset.

**5.2.3. Goal Inference.** In our multiagent joint goal inference algorithm based on SIS PF with resampling, we set particle number  $N_p$  according to experiment needs. We also make the resampling threshold  $N_T$  equal to one-third of the particle number  $N_p$ .

**5.3. Experiment Results and Discussion.** The paper first retrieves the agents' optimal policies using MARL algorithm. Based on that, we run the agent decision model repeatedly and collect a test dataset consisting of 100 labeled traces. After analyzing the dataset, we find that there are on average 28.05 steps in one trace, and the number of steps in one trace varies from 16 to 48, respectively, with a standard deviation of 9.24. Also we find that among 100 traces, there are approximately

TABLE 1: The details of two traces.

Trace Number	Durations	Targets	Goal interrupted
1	$t \in [1, 22]$	Prey B	Yes
	$t \in [23, 28]$	Prey A	No
13	$t \in [1, 22]$	Prey B	No

60% traces where predators changed their joint goal for at least once halfway, 27% where goals are changed at least twice, and 15% where goals changed greater than or equal to three times. The statistics above almost cover all situations we need in validation of our method.

Based on the test dataset, we did our experiments on three aspects: (a) to discuss details of the multiagent goal recognition, present and analyze results of two specific traces, and testify to the ability of our method in recognizing dynamic changing goals; (b) to compare the performance of joint goal recognition under Dec-POMDM-T framework and that of Dec-POMDM [9] in terms of precision, recall, and  $F$ -measure; (c) to show the effectiveness of our multiagent goal inference method based on SIS PF with resampling.

**5.3.1. Goal Recognition of Specific Traces.** To show the details of the recognition results, we select two specific traces from the dataset (Trace Number 1 and Number 13). These two traces are selected because Trace Number 1 is the first trace where the goal is changed before it is achieved, while Number 13 is the first trace where the goal is kept until it is finally achieved. The detailed information is shown in Table 1.

Given the optimal policies and other parameters of the Dec-POMDM-T including  $I, S, A, G, \Omega, Y$ , and  $h$ , we used the

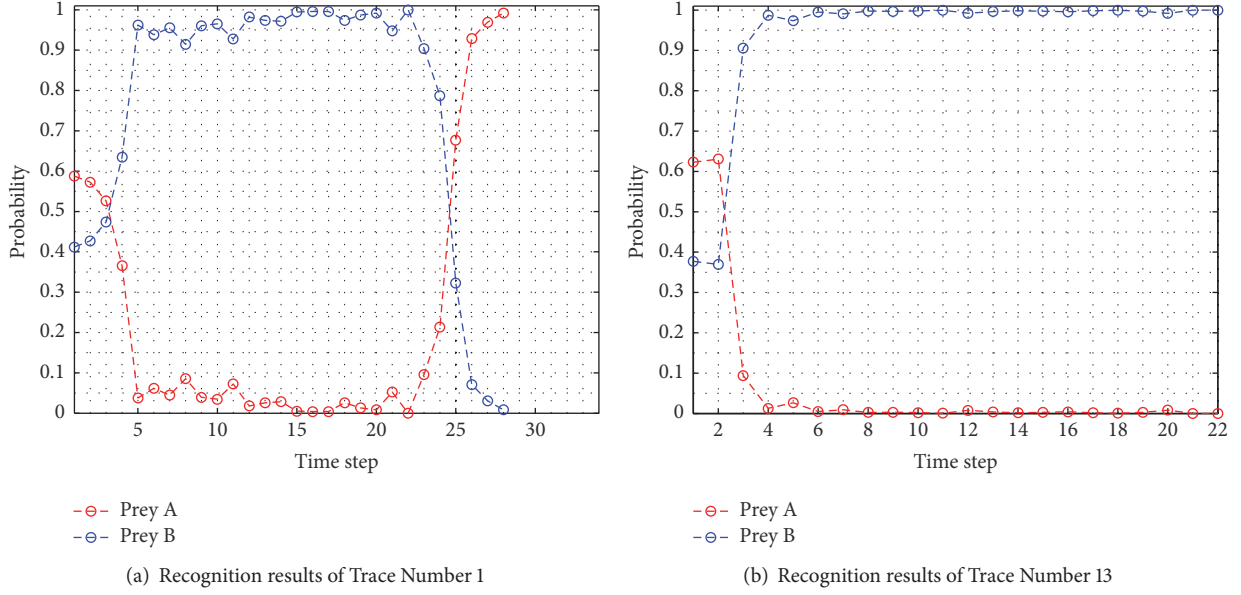


FIGURE 7: Recognition results of two specific traces under the Dec-POMDM-T.

SIS PF with resampling to compute the posterior distribution of goals at each time. In Trace Number 1, predators first selected Prey B as their joint goal from  $t = 1$  to  $t = 22$ . As the initial distribution of goal Prey A and Prey B was set to be 0.6 and 0.4, the blue line which represents the probability of agents pursuing Prey B started from 0.4. It then rose up as more evidence came in and finally overran the red line at  $t = 4$ . This trend continued with occasional bumps until predators changed their goals at  $t = 23$ . As the joint goal had been changed to Prey A, the red line reacted fast between time step  $t = 24$  and  $t = 25$ . Finally, agents achieved their goal at  $t = 28$ . Trace Number 1 proves the effectiveness of our method in recognizing dynamic changing goals.

In Trace Number 13, predators selected Prey B as their initial goal. The goal was kept until it was achieved at  $t = 22$ . From Figure 7(b) we can see that, our method reacted very fast to observation information, and the probability of Prey B as the joint goal rose directly from no more than 0.4 towards 0.9 at  $t = 3$ . This high confidence continued and stayed at almost 1 along the whole recognition process. Besides, the algorithm in Figure 7(b) shows its ability in reaching early convergence point for multiagent joint goal recognition.

### 5.3.2. Comparison of the Dec-POMDM-T and Dec-POMDM.

As stated above, the performance comparisons are made in terms of three classic metrics in goal recognition domain, which are precision, recall, and  $F$ -measure [36]. They are computed as

$$\text{precision} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TI_i}$$

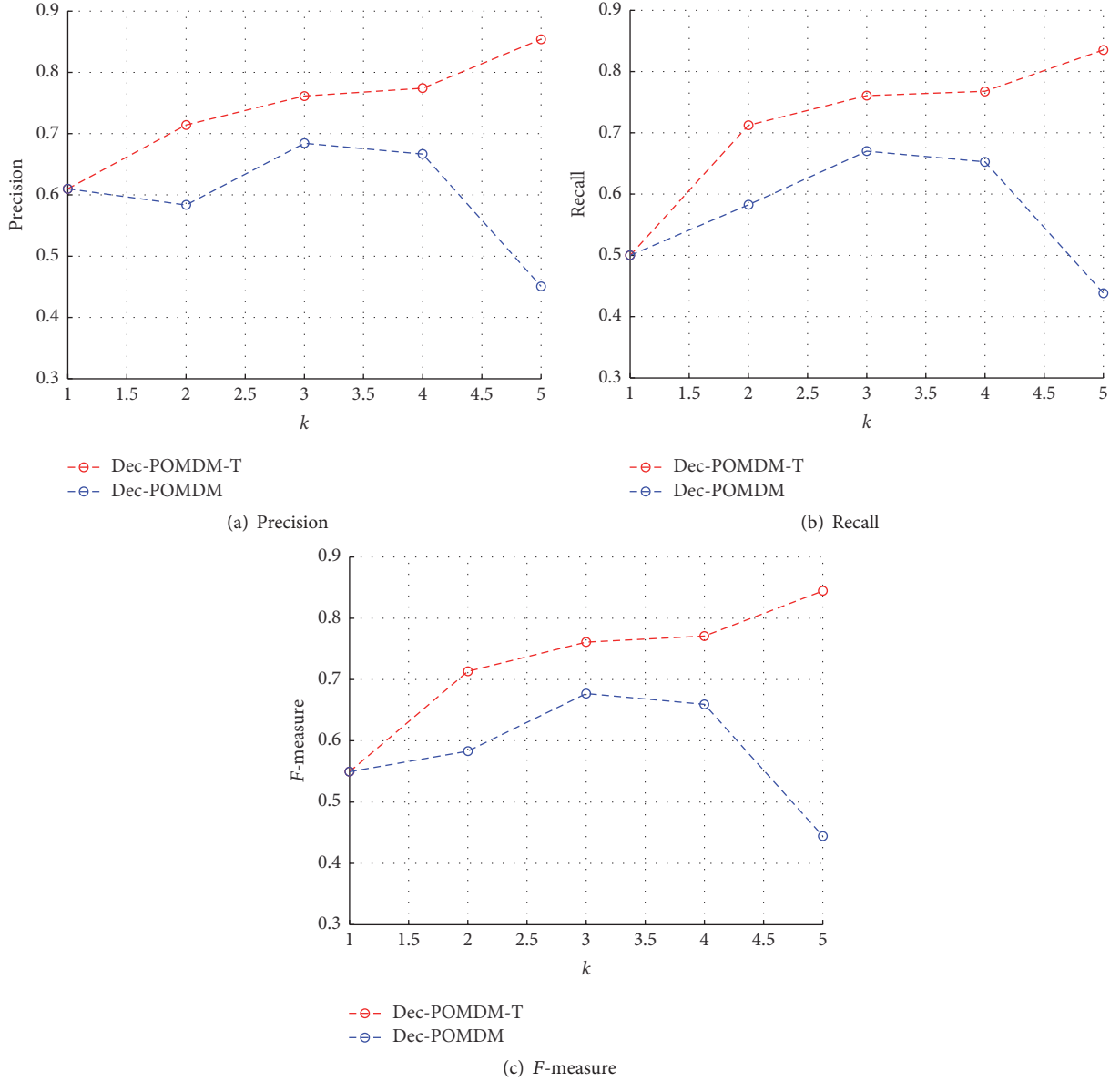
$$\text{recall} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TT_i}$$

$$F\text{-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \quad (8)$$

where  $N$  is the number of possible goals.  $TP_i$ ,  $TI_i$ , and  $TT_i$  are the true positives, total of true labels, and total of inferred labels for class  $i$ , respectively. Formulas (8) show that, precision is used to scale the reliability of the recognized results; recall is used to scale the efficiency of the algorithm applied in the test data set; and  $F$ -measure is an integration of precision and recall. We can find that the value of all these metrics will be between 0 and 1, and a higher metric means a better performance. In order to solve the problem of traces having different lengths, the paper defines a positive integer  $k$  ( $k = 1, 2, \dots, 5$ ). The corresponding observation sequences are  $\{y_{t \in 1:100}^{j \in 1:100}\}_{t \in 1:[k \cdot \text{length}^j/5]}$ . Here,  $y_{t \in 1:[k \cdot \text{length}^j/5]}^j$  is the observation sequence from time 1 to time  $[k \cdot \text{length}^j/5]$  of the  $j$ th trace; the  $\text{length}^j$  is the length of the  $j$ th trace. The metrics under different  $k$  show the models' performance in different simulation phases.

It is obvious in Figure 8 that the performance of Dec-POMDM-T was much better than Dec-POMDM when more observations were received. Specifically, all the three metrics of the Dec-POMDM-T had exceeded 0.75 when more than half of traces had been observed at  $k \geq 3$ . The Dec-POMDM, however, did not perform that well in all three. This is mainly because Dec-POMDM has no definition of action durations. As predators will not select actions in every time step, the filtering process of Dec-POMDM would usually fail.

**5.3.3. Effectiveness of Multiagent Goal Inference Based on SIS PF with Resampling.** In this section, we test the effectiveness of our multiagent goal inference based on SIS PF with resampling. In Figure 9, we first give the changing patterns

FIGURE 8: Three metrics of Dec-POMDM-T and Dec-POMDM ( $N_p = 6000$ ).

of variances for above-mentioned two specific traces. The weighted variances at time  $t$  are computed by

$$\text{Var}_t = \sum_{i=1}^n w_t^i (g_t^i - \hat{g}_t) (g_t^i - \hat{g}_t)^T, \quad (9)$$

where  $w_t^i$  is the weight of particle  $x_t^i$  and  $\hat{g}_t$  is the estimated goal distribution in  $x_t^i$ . From Figure 9, it is obvious that all variances of two traces had large values at the beginning and they would all be affected by noisy observations or observations containing vague information. Then they dropped with more information coming in. The variance for Trace Number 13 in Figure 9(b) dropped continually along the recognition process with several small up and downs as of reasons above. Similar situations happened in Trace Number

1 in Figure 9(a). However, its variance rose up dramatically when agents changed their joint goal halfway. This happened at  $t = 23$ , as shown in Table 1, and thus pushed up the variance to more than 0.4. Finally, the curve dropped down fast to less than 0.05 within 3 time steps and now the estimated goal was changed from Prey B to Prey A.

We also conduct experiments on variance using goal inference algorithm with different particle numbers. The difference between the red and blue lines is that the former exploits 4000 particles while the latter 8000. The results show that variances are not sensitive to the particle number of PF algorithm. It can get good performance with a few particles.

As a common problem in PF algorithms, particles may not survive till the end of goal recognition process as their number is not enough. In this scenario when  $N_p \leq 1000$ ,

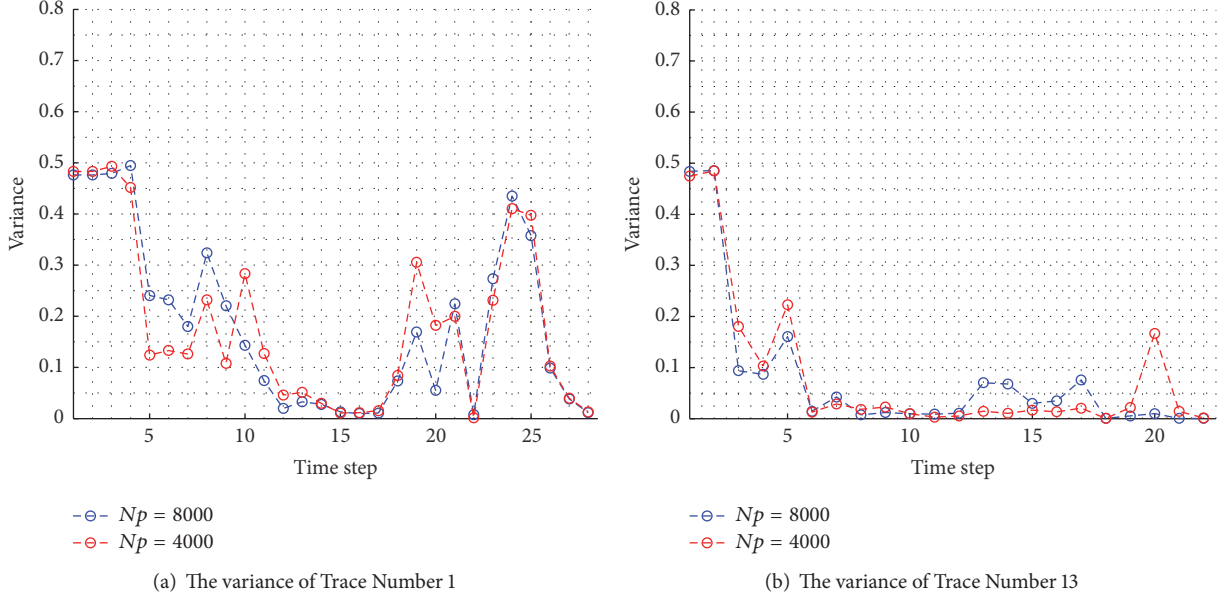
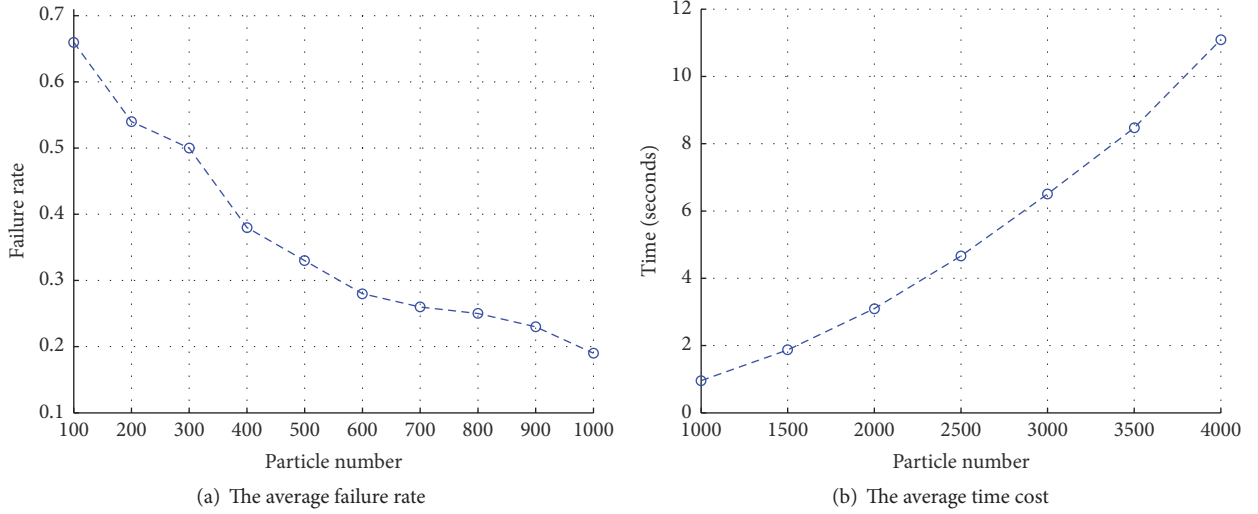
FIGURE 9: Variances of Traces Number 1 and Number 13 ( $N_p = 4000, 8000$ ).

FIGURE 10: The average failure rate and time cost with different numbers of particles.

the goal inference algorithm may suffer from serious failure. To view the specific effects of it, we ran the test dataset for 10 times with different numbers of particles. The average failure rates are shown in Figure 10(a) and also summarized in Table 2. Two more rates when the numbers of particles is equal to 4000 and 6000 are also given in Table 2. Obviously, average failure rate drops significantly as the particle number gets larger.

The time cost with different particle numbers is shown in Figure 10(b). The program was written in Matlab script and ran in computer with an Intel Core i7-4770 CPU (3.40 GHz). We can see that time cost would increase as we expand

particle population. Consider the considerably long effects of agent intention; this approximate inference method would still be applicable under certain combination of parameter settings. Further, we also compare the precision, recall, and  $F$ -measure under different number of particles as in Figure 11.

In Figure 11, the red, blue, cyan, green, and magenta dashed curves indicate the metrics of SIS PF with resampling each with 1000, 2000, 4000, 6000, and 16000 particles, respectively. The PF with the largest particle number, having all metrics reaching to almost 0.9 at last, performed best than the remaining ones. Filters with numbers 4000 and 6000 had similar trends along the process and came even closer at



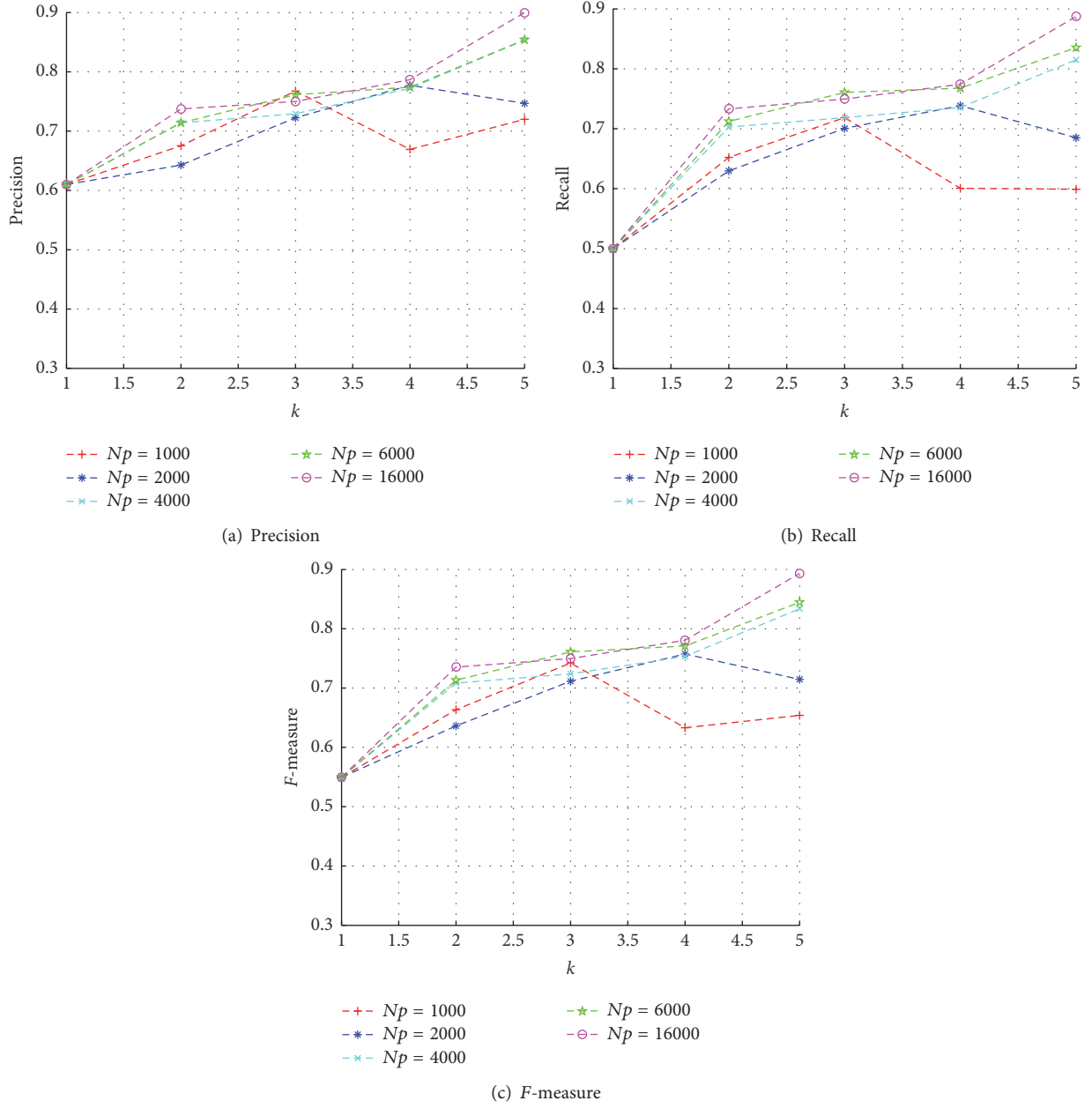


FIGURE 11: Three metrics of the recognition results with different particle numbers.

last, while filters with numbers 1000 and 2000 performed the worst as of being short of particles.

## 6. Conclusions

In this paper, we propose a novel model for solving multi-agent goal recognition problems, the Dec-POMDM-T, and present its corresponding learning and inference algorithms, which solve a multiagent goal recognition problem. First, we use the Dec-POMDM-T to model the general multiagent goal recognition problem. The Dec-POMDM-T presents the agents' cooperative behaviors in a compact way, and thus the cooperation details are unnecessary in the modeling process.

It can also make use of existing algorithms for solving the Dec-POMDP problem. Then we use the SIS particle filter with resampling to infer goals under the framework of the Dec-POMDM-T. Last, we also design a modified predator-prey problem to test our method. In this modified problem, there are multiple possible joint goals and agents may change their goals before they are achieved. Experiment results show that (a) Dec-POMDM-T works effectively in multiagent goal recognition and adapts well to dynamic changing goals within agent group; (b) Dec-POMDM-T outperforms traditional Dec-MDP-based methods in terms of precision, recall, and F-measure. In the future, we plan to apply the Dec-POMDM-T in more complex scenarios.

TABLE 2: The average failure rates with different numbers of particles.

Particle numbers	Average failure rates
100	66%
200	54%
300	50%
400	38%
500	33%
600	28%
700	26%
800	25%
900	23%
1000	19%
4000	8%
6000	2%

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

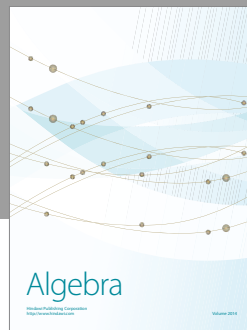
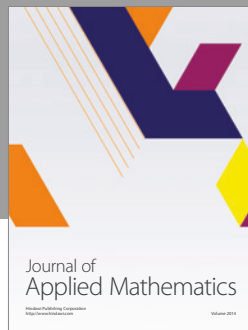
## Acknowledgments

This work is sponsored by the National Natural Science Foundation of China under Grant no. 61473300.

## References

- [1] S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game AI research and competition in starcraft," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5, no. 4, pp. 293–311, 2013.
- [2] S. C. J. Bakkes, P. H. M. Spronck, and G. van Lankveld, "Player behavioural modelling for video games," *Entertainment Computing*, vol. 3, no. 3, pp. 71–79, 2012.
- [3] D. Churchill, D. Churchill, Sparcraft: open source StarCraft combat simulation, 2013, <http://code.google.com/p/sparcraft/>.
- [4] H. H. Bui, S. Venkatesh, and G. West, "Policy recognition in the abstract hidden Markov model," *Journal of Artificial Intelligence Research*, vol. 17, pp. 451–499, 2002.
- [5] A. Hoogs and A. A. Perera, "Video activity recognition in the real world," in *Proceedings of the 23rd National Conference on Artificial Intelligence*, pp. 1551–1554, Chicago, Ill, USA, July, 2008.
- [6] C. L. Baker, R. Saxe, and J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.
- [7] K. Yordanova, F. Krüger, and T. Kirste, "Context aware approach for activity recognition based on precondition-effect rules," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops '12)*, pp. 602–607, IEEE, Lugano, Switzerland, March 2012.
- [8] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [9] S. Yue, K. Yordanova, F. Krüger, T. Kirste, and Y. Zha, "A decentralized partially observable decision model for recognizing the multiagent goal in simulation systems," *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 5323121, 15 pages, 2016.
- [10] M. Baykal-Gürsoy, *Semi-Markov Decision Processes*, Wiley Encyclopedia of Operations Research and Management Science, 2010.
- [11] M. Ramirez and H. Geffner, "Goal recognition over POMDPs: inferring the intention of a POMDP agent," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, pp. 2009–2014, Barcelona, Spain, July 2011.
- [12] B. Scherrer and F. Charpillet, "Cooperative co-learning: a model-based approach for solving multi agent reinforcement problems," in *Proceedings of the 14th International Conference on Tools with Artificial Intelligence*, pp. 463–468, November 2002.
- [13] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT press, 2009.
- [14] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.
- [15] N. Oliver, E. Horvitz, and A. Garg, "Layered representations for human activity recognition," in *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, pp. 3–8, IEEE, Pittsburgh, Pa, USA, 2002.
- [16] L. Liao, D. Fox, and H. Kautz, "Hierarchical conditional random fields for GPS-based activity recognition [C]," in *Proceedings of the International Symposium of Robotics Research*, 2005.
- [17] S. Hladky and V. Bulitko, "An evaluation of models for predicting opponent positions in first-person shooter video games," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG '08)*, pp. 39–46, IEEE, Perth, Australia, December 2008.
- [18] T. Duong, D. Phung, H. Bui, and S. Venkatesh, "Efficient duration and hierarchical modeling for human activity recognition," *Artificial Intelligence*, vol. 173, no. 7–8, pp. 830–856, 2009.
- [19] L. Getoor and B. Taskar, Eds., *Introduction to Statistical Relational Learning*, MIT Press, 2007.
- [20] K. Kersting, L. De Raedt, and T. Raiko, "Logical hidden Markov models," *Journal of Artificial Intelligence Research*, vol. 25, pp. 425–456, 2006.
- [21] S. Raghavan, P. Singla, and R. J. Mooney, "Plan recognition using statistical-relational models," in *Plan, Activity, and Intent Recognition: Theory and Practice*, G. Sukthankar, R. P. Goldman, C. Geib, D. V. Pynadath, and H. H. Bui, Eds., Morgan Kaufmann Publishers, Waltham, MA, USA, 2014.
- [22] K. Kersting and L. De Raedt, "Towards combining inductive logic programming with Bayesian networks," in *Proceedings of the 11th International Conference on Inductive Logic Programming*, pp. 118–131, 2001.
- [23] C. W. Geib and M. Steedman, "On natural language processing and plan recognition," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007*, pp. 1612–1617, January 2007.
- [24] W. Min, E. Y. Ha, J. Rowe, B. Mott, and J. Lester, "Deep learning-based goal recognition in open-ended digital games," in *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '14)*, pp. 37–43, Raleigh, NC, USA, October 2014.
- [25] S. Keren, A. Gal, and E. Karpas, "Goal recognition design with non-observable actions," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 3152–3158, February 2016.

- [26] S. Keren, A. Gal, and E. Karpas, "Goal recognition design for non-optimal agents," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015*, pp. 3298–3304, January 2015.
- [27] S. Saria and S. Mahadevan, "Probabilistic plan recognition in multiagent systems," in *Proceedings of the 14th International Conference on Automated Planning and Scheduling, ICAPS 2004*, pp. 287–296, June 2004.
- [28] J. Yin, D. H. Hu, and Q. Yang, "Spatio-temporal event detection using dynamic conditional random fields[C]," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 1321–1327, 2009.
- [29] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction[C]," in *Proceedings of the 17th Annual Conference Neural Information Processing Systems*, pp. 1185–1192, 2004.
- [30] T. T. Truyen, D. Q. Phung, H. H. Bui, and S. Venkatesh, "Hierarchical semi-Markov conditional random fields for recursive sequential data[C]," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems, NIPS 2008*, pp. 1657–1664, December 2008.
- [31] T. D. Ullman, C. L. Baker, O. Macindoe, O. Evans, N. D. Goodman, and J. B. Tenenbaum, "Help or hinder: bayesian models of social goal inference," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 1874–1882, December 2009.
- [32] B. Riordan, S. Brimi, N. Schurr et al., "Inferring user intent with Bayesian inverse planning: making sense of multi-UAS mission management," in *Proceedings of the 20th Annual Conference on Behavior Representation in Modeling and Simulation (BRiMS '11)*, pp. 49–56, Sundance, Utah, USA, March 2011.
- [33] Q. Yin, S. Yue, Y. Zha, and P. Jiao, "A semi-Markov decision model for recognizing the destination of a maneuvering agent in real time strategy games," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1907971, 12 pages, 2016.
- [34] G. E. Monahan, "State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms," *Management Science*, vol. 28, no. 1, pp. 1–16, 1982.
- [35] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [36] G. Sukthankar, C. Geib, H. H. Bui, D. Pynadath, D. Pynadath, and R. P. Goldman, Eds., *Plan, Activity, and Intent Recognition: Theory and Practice*, Newnes, 2014.



Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

