# Inverse Reinforcement Learning based Human Behavior Modeling for Goal Recognition in Dynamic Local Network Interdiction*

**Yunxiu Zeng[1], Kai Xu[1], Quanjun Yin[1], Long Qin[1], Yabing Zha[1], William Yeoh[2]**
1. The Institute of Simulation Engineering,
College of Information System and Management, NUDT, Changsha, 410073, China.
{zengyunxiu12@, xukai09@, yinquanjun@, qinlong@, zhayabing@}nudt.edu.cn
2. Department of Computer Science and Engineering,
Washington University in St. Louis, St. Louis, MO 63130, USA.
wyeoh@wustl.edu

## Abstract

Goal recognition is the task of inferring an agent's goals given some or all of the agent's observed actions. Among different ways of problem formulation, goal recognition can be solved as a model-based planning problem using off-the-shell planners. However, obtaining accurate cost or reward models of an agent and incorporating them into the planning model becomes an issue in real applications. Towards this end, we propose an *Inverse Reinforcement Learning* (IRL)-based opponent behavior modeling method, and apply it in the goal recognition assisted Dynamic Local Network Interdiction (DLNI) problem. We first introduce the overall framework and the DLNI problem domain of our work. After that, an IRL-based human behavior modeling method and *Markov Decision Process*-based goal recognition are introduced. Experimental results indicate that our learned behavior model has a higher tracking accuracy and yields better interdiction outcomes than other models.

## Introduction

The ability to recognize the plans and goals of other agents enables humans, AI agents or command-and-control systems to reason about what the other agents are doing, why they are doing it, and what they will do next (Sukthankar *et al.* 2014). Plan and goal recognition systems work well in many applications like human-robot interaction (Hofmann and Williams 2007), dialogue understanding (Litman and Allen 1987) and system intrusion detection (Geib and Goldman 2001). They have been formulated and addressed in many ways, as a matching problem over a suitable AND/OR graph (Avrahami-Zilberbrand and Kaminka 2005), a parsing problem over grammar (Pynadath and Wellman 1998), a probabilistic inference task over a dynamic Bayesian network (Bui *et al.* 2002; Liao *et al.* 2007) and an inverse planning problem over planning models (Baker *et al.* 2009; Ramırez and Geffner 2011). All the above methods have to leverage on the domain knowledge to some extent, for example in the form of predefined libraries of plans or policies, or a pre-specified agent action model along with a set of possible goals. However, there is little research on plan or goal recognition that focuses on how to retrieve this information from the real world.

In this work, we present a framework of using offline-learned and online-modified opponent behavior model in *Markov Decision Process* (MDP)-based goal recognition, and further apply the goal recognizer in the Dynamic Local Network Interdiction (DLNI) problem to interdict the observed agent. Inverse reinforcement learning is used to learn the agent-specific action model offline, i.e. a feature-based reward model, from sets of example traces that are collected from the same group of subjects. The model, modified by online environment changes using reinforcement learning, then replaces the original decision model (e.g. a path planner) in an approximate inference algorithm (e.g. particle filter) in the MDP-based goal recognition module. Network interdiction is a classic Operations Research problem applied in domains involving critical infrastructure protection (Scaparra and Church 2008), public transportation (Laporte *et al.* 2010) and public security (Cappanera and Scaparra 2011). The work in (Xu *et al.* 2017) first relaxes the original "static and pre-known goal" assumption and orients the knowledge generated by goal recognition system into the decision-making process of the interdictor. However, their work does not learn the evader behavior from data and thus neglects the existence of preference and variability in human behaviors.

Our results show that the learned opponent behavior model has higher tracking accuracy and results in more effective network interdiction than other models. Besides, IRL method also gives better generalization ability to the behavior model as its feature-based reward structure could be extended to areas the evader has not visited or other maps.

## Background and Related Work

### IRL-based Human Behavior Modeling

Machine learning methods, including reinforcement learning (RL) (Yue *et al.* 2016), deep learning (Min *et al.* 2014; Bisson *et al.* 2015) and IRL (Tastan *et al.* 2012), have already been applied in learning the agents' behavior models for goal and plan recognition. Among these approaches, RL and IRL both focus on the reward or cost of agent behaviors, and have better generalization ability compared with supervised or unsupervised learning methods. Compared to

*Yunxiu Zeng and Kai Xu are both first authors of this paper.

RL whose reward function has to be designated manually by human experts, IRL further relaxes the existence of this function and takes the preference and variability of human behaviors into consideration. The most notable series of applications of IRL is the autonomous helicopter aerobatic demonstrations (Abbeel *et al.* 2007), as well as automated car driving simulator that can learn different driving styles (Abbeel and Ng 2004). Recently, IRL has also been applied in human behavior learning and modeling, as in the pedestrian intentions prediction in a robot application (Ng *et al.* 2000) and a user simulation database for dialogue (Chandramohan *et al.* 2011). Tasten *et al.* (Tastan *et al.* 2012) also use IRL to learn opponents behavior and intercept them in first person shooter games. In this paper, the Maximum Entropy IRL (Ziebart *et al.* 2008), a feature-based model is used to learn the opponent moving behavior.

## Goal Recognition and Dynamic Local Network Interdiction

As we have discussed above, plan or goal recognition problem could be formulated and addressed in many ways. Among these approaches, two formulations solve the problem from different perspectives. One uses a pre-defined library of plans or policies; the other one replaces it by an agent action model and a set of possible goals (Ramırez and Geffner 2011). Apart from advantages including easy use of existing off-the-shell planners, the latter one is also much more flexible and expressive in representing complex low-level agent behaviors, like shooting and moving. In most model-based researches (Baker *et al.* 2009; Ramırez and Geffner 2011; Yin *et al.* 2016), posterior goal distribution $P(G|O)$ is usually obtained from the Bayes rule $P(G|O) = \alpha P(O|G)P(G)$, where $\alpha$ is a normalizing constant. This is also applicable for models using probabilistic goal inference method (Yue *et al.* 2016).

The network interdiction problem has been examined for several decades within the context of a variety of modeling approaches, optimization objectives, and solution techniques. The network interdiction problem that we focus on is frequently referred to as the Shortest Path Network Interdiction (SPNI) problem. SPNI is the interdictor's problem: subject to a limited interdiction budget, interdict arcs in a network to maximize the shortest path length between specified nodes $s$ and $t$ (Israeli and Wood 2002). It could be viewed as a bilevel mixed-integer program (BLMIP), which is a special case of a static Stackelberg game (Simaan and Cruz 1973). Previous network interdiction researches usually neglect the fact that, in real-life scenarios, the evader would change its goal stochastically, or even deceptively. Using online goal recognition, the work in (Xu *et al.* 2017) first relaxes this assumption and studies a dynamic local network interdiction problem. However, the evader's behavior model that they use during the goal inference only applies a simple distance-base heuristic method, and thus cannot accurately reflect the characteristics of human behaviors.

# Methodology

As in Figure 1, the two blue blocks MDP-based goal recognition and DLNI both formulate the second $o$ and the third $d$ of *observe-orient-decide-act* loop, where the *observe* and *act* are abbreviated as "Observed Action Sequence" and "Interdicted Arcs" and depicted using blue lines. The learning process is to calculate the policy, which defines the probabilistic action selection in every state for agents marching for different destinations, from the opponent behavior datasets in the form of moving trajectories. Firstly, these trajectories would be preprocessed and separated into sets of fragments according to different goals, in case of goal midway changes. Then maximum entropy IRL learns the opponent behavior model through weight estimation of predefined behavior features, through calculating the opponent feature count $\mathbf{f}_{evader}$, the number of times features were observed in the set of trajectories. After that, reinforcement learning algorithm would be further used to retrieve the optimal policy from the reward function, which is a linear form of features and weights. Till now, all learning steps are processed off-line. In the runtime, the opponent behavior model would be further modified online considering environmental changes caused by arcs interdiction.

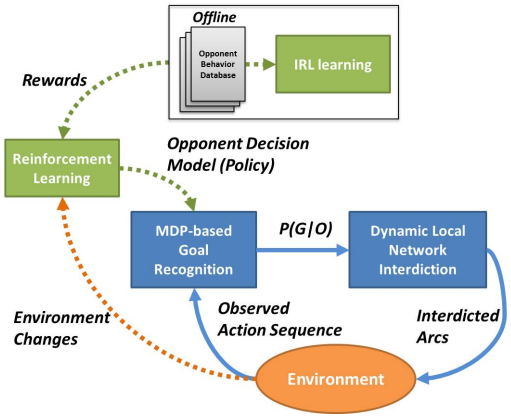



Figure 1: The Framework of Inverse Reinforcement Learning based Human Behavior Modeling for Goal Recognition in Dynamic Local Network Interdiction

First and foremost, we give the problem formulation of DLNI (Xu *et al.* 2017) as follows.

**Problem:** Maximize the expectation shortest $s - g$ path length in a directed network by interdicting arcs,

**Indices:** $i \in N$, nodes in $G$ ($s$ is the current source node, $g_1, \ldots, g_m$ are the potential termini),
$k = (i, j) \in A$, arcs in $G$,
$k \in FS(i)(k \in RS(i))$, arcs directed out of (into) node $i$,
$\tau = 1, 2, \ldots, T$, stages of the confrontation process,

**Data:** $0 \leq c_k < \infty$, nominal length of arc $k$ (vector form $\mathbf{c}$),

$0 \leq d_k < \infty$, added integer delay if arc $k$ is interdicted (vector form $\mathbf{d}$),
$r_k > 0$, resource required to interdict arc $k$ (vector form $\mathbf{r}$),
$R$, total amount of interdiction resource,
$R_\tau$, total amount of interdiction resource assigned to stage $\tau$,
$0 \leq p(g_j) < 1, \sum_{j=1,\cdots,m} p(g_j) = 1$, the probabilistic distribution over the possible goals $g_1, \cdots, g_m$,

**Variables:** $x_k = 1$ if arc $k$ is interdicted by the interdictor; else $x_k = 0$,
$y_k = 1$ if arc $k$ is traversed by the evader; else $y_k = 0$

The formulations is:

$$\textbf{[DSPLNI-P]} \quad \max_{\mathbf{x} \in X} \min_{\mathbf{y}} \sum_{k \in A} (c_k + x_k d_k) y_k$$

$$\sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k = \begin{cases} 1 & \text{for} \quad i = s \\ 0 & \forall i \in N \backslash \{s, g_1, \cdots, g_m\} \\ -p(g_j) & \forall i = g_j, j \in \{1, \cdots, m\} \end{cases} \tag{1}$$

$$x_k \in \{0, 1\}, \quad \forall k \in FS(s) \tag{2}$$

$$x_k = 0, \quad \forall k \notin FS(s) \tag{3}$$

$$y_k \geq 0, \quad \forall k \in A \tag{4}$$

where $X = \{\mathbf{x} \in \{0,1\}^{|A|} | \mathbf{r}^T \mathbf{x} \leq R_\tau\}$.

## IRL-based Human Behavior Modeling

Taking dynamically changing goals into consideration, our goal-MDP model is defined as a tuple $\langle s_0, S, G, e, A, P_a(s'|s), O \rangle$, where $s_0$ is the initial state, $S$ is the finite set of states, $G \subseteq S$ is the non-empty set of goal states, $e = \{0, 1\}$ is the goal termination variable for $e = \{0, 1\}$, $A$ is the set of actions associated with each state, $P_a(s'|s)$ is the action selection probability with $a \in A$, $s, s' \in S$, and $O$ is the non-empty observation set. Essentially, the model is a dynamic Bayesian network (DBN), in which all causalities could be depicted. We present a full DBN structure depicting two time slices in Figure 2.

The learning process of IRL is based on a cycle of: 1) forward-backward passes to calculate the expected feature counts for a given set of weights, and 2) gradient-based feature weight updates. Our method starts with collecting observation data from a human opponent and separating trajectories into several fragments by the goals, if the goal changes in a trajectory. Given all the observed data, the goal is to find the policy $\pi$ that best matches the data under the maximum entropy assumption.

Note that the policy depends on the reward at each step, and the reward is assumed to be the linear of features multiplied by a weight vector: $R(\mathbf{f}, \vec{\omega}) = \vec{\omega} \cdot \mathbf{f}$. In this task we use the following features:

- distance $d$ to the evader's goal;
- the value of coordinate $X$; and
- the value of coordinate $Y$.

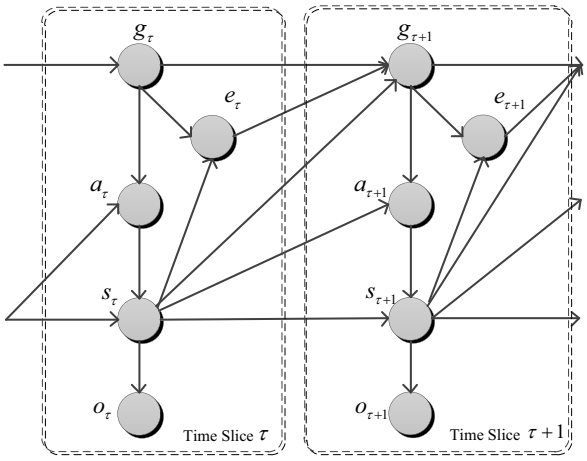



Figure 2: The DBN structure of the model

Thus, $R = \omega_1 \times d + \omega_2 \times X + \omega_3 \times Y$, where $\vec{\omega} = (\omega_1, \omega_2, \omega_3)$ are the weights of three features.

The opponent's policy $\pi$ is defined as $\pi_{\vec{\omega}} := P(a|s, \vec{\omega})$, where $\vec{\omega}$ is given by $arg\max_{\vec{\omega}} L(\vec{\omega}) = \sum_{m=1}^{M} \log P(traj_m | \vec{\omega})$. Fortunately, this function is convex for deterministic MDPs (such as game worlds) and therefore can be solved using gradient descent:

$$dL(\vec{\omega}) = E[\mathbf{f}_{evader}] - E[\mathbf{f}_{\pi_{\vec{\omega}}}] \tag{5}$$

$E[\mathbf{f}_{evader}]$ is straightforward from observation data and assumes $C_{s_i}$ is the expected visiting count for $s_i$.

$$E[\mathbf{f}_{\pi_{\vec{\omega}}}] = \sum_{s_i} C_{s_i} \mathbf{f}_{s_i} | \pi_{\vec{\omega}} \tag{6}$$

The forward-backward pass algorithm for computing expected occupancy counts based on weights can be found in (Ziebart *et al.* 2008). With this information we can compute the expected feature count of the whole policy using Eq. (6). The gradient descent method (Eq. (5)) is used to improve $\vec{\omega}$ until it reaches its termination criteria.

The policy can be retrieved by calculating the value for each state $s$ at time $t$ using the Bellman equation:

$$V(s_t) = \max_{a_{s_t}} \{R(s_t, a_{s_t}) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_{s_t}) V(s_{t+1})\} \tag{7}$$

The Bellman equation formulation for value iteration presented above would give us a fixed sequence of actions for each state that achieves the maximum value for the MDP. A similar *state-action value function* $Q : S \times A \rightarrow R$, can be defined as the expected return starting from state $s$, taking action $a$ and thereafter following policy:

$$Q(s, a) = \sum_{s'} P(s'|s_t, a_{s_t})(R(s, a, s') + V(s')) \tag{8}$$

Rather than simply choosing the maximum, humans exhibit flexibility when choosing the actions, especially when several options are all strong. To accommodate this behavior, we assume that the opponent selects from a distribution of actions in each state. The distribution with maximum information entropy will minimize the information

loss and thus result in more human-like behavior. Thus, rather than just selecting the action with the maximum reward, the probability of an action can be weighted by the expected exponential values $P(a_j|s_i) \propto \exp Q(s_i, a_j)$. The corresponding Bellman equation assuming that the opponent draws from distribution of actions is given in Eq. (8) where $R(s,a) = \vec{\omega} \cdot \mathbf{f}$.

The process of calculating reward function using Maximum Entropy IRL is shown in Algorithm 1. After $\vec{\omega}$ is initialized randomly, the $\mathbf{f}_g$ is updated according to current state $s$ and goal $g$. Then the features are added up from behavior trajectories. Finally, $\vec{\omega}$ could be computed using the forward-backward pass algorithm and gradient-based feature weight updates, as in Eq. (5).

**Algorithm 1:** Policy Calculation under Different Goals by Maximum Entropy IRL

```
Initialize ω⃗^0 ← (ω_1, ω_2, ω_3);
foreach g ∈ G do
    f_g = FeatureUnderTarget(g);
    E[f_evader] = Σ_{m=1}^{M} f_{traj_m};
    ω⃗_g = MaxEntIRL(f_evader, f_g, ω⃗^0);
    R_g(f_g, ω⃗_g) = ω⃗_g · f_g
end
```

## Dynamic Goal Recognition in DLNI

Recognizing the evader's goal is an inference problem trying to find the real goal behind agent actions based on observations online. In essence, the task is to compute the posterior distribution $P(g_\tau|o_\tau)$ of goal $g_\tau$ given observation $o_\tau$. This could be achieved either by accurate inference or by approximate methods. Widely applied in sequential state estimation, particle filter is a kind of approximate inference methods designed to handle non-Gaussian, nonlinear and high-dimensional problems (Chen and others 2003). In this work, the set of possible goals is given along with the priors $P(G)$. Similar assumptions also exist in (Ramırez and Geffner 2011) in which the posterior goal probabilities $P(G|O)$ is obtained from Bayes rule $P(G|O) = \alpha P(O|G)P(G)$ where $\alpha$ is a normalizing constant. In particle filter however, a posterior distribution is empirically represented using a weighted sum of $N_p$ samples (Chen and others 2003) drawn from the proposal distribution:

$$p(g_\tau|o_\tau) \approx \sum_{i=1}^{N_p} W_\tau^{(i)} \delta(g_\tau - g_\tau^{(i)}) \tag{9}$$

where $g_\tau^{(i)}$ are assumed to be *i.i.d* drawn from $q(g_\tau|o_i)$. The importance weights $W_\tau^{(i)}$ should be updated recursively

$$W_\tau^{(i)} \approx W_{\tau-1}^{(i)} \frac{p(o_\tau|g_\tau^{(i)}) p(g_\tau^{(i)}|g_{\tau-1}^{(i)})}{q(g_\tau^{(i)}|g_{0:\tau-1}^{(i)}, o_\tau)} \tag{10}$$

As we use simplest sampling, the $q(g_\tau^{(i)}|g_{0:\tau-1}^{(i)}, o_\tau)$ is set to be $p(g_\tau^{(i)}|g_{\tau-1}^{(i)})$, which could be computed directly using the agent action model. Thus the $g_\tau$ in Eq. (9) would be sampled from $p(g_\tau^{(i)}|g_{\tau-1}^{(i)})$. As the observation $o_\tau$ only depends on $s_\tau$, the importance weights $W_\tau^{(i)}$ can be updated by

$$W_\tau^{(i)} = W_{\tau-1}^{(i)} \cdot p(o_\tau|s_\tau^{(i)}). \tag{11}$$

The process of applying IRL-assisted dynamic goal recognition in DLNI at one confrontation stage is shown in Algorithm 2. Variables including $Q$ value, action selection $Policy$, the adjacent matrix $Adj$ describing current state in a form of vertexes and arcs, feature value $\mathbf{f}$ and particle system $Particle$ serve as inputs at stage $\tau - 1$ and outputs at $\tau$. In line 3 of Algorithm 2, after observing the evader's action at $\tau$ from environment, the goal recognition module uses the $Policy_{\tau-1}$ computed at stage $\tau - 1$ to advance the particle system from $Particle_{\tau-1}$ to $Particle_\tau$. The weights are updated using Eq. (11) by comparing the current states of particles in $Particle_\tau$ and the real observation $Obs$.

**Algorithm 2:** IRL-assisted Goal Recognition in DLNI at confrontation stage $\tau$

**Input**: $Q_{\tau-1}, Policy_{\tau-1}, Adj_{\tau-1}, \mathbf{f}_{\tau-1}, Env_\tau, Particle_{\tau-1}$
**Output**: $Q_\tau, Arcs, Adj_\tau, Policy_\tau, \mathbf{f}_\tau, Particle_\tau$

1 $\tau = \tau + 1$;
2 $Obs = observe(Env_\tau)$;
3 $(Particle_\tau, P(G|O)) = goalRecognition(Particle_{\tau-1}, Policy_{\tau-1}, Obs)$;
4 $Arcs = DLNI(Adj_{\tau-1}, P(G|O))$;
5 $(Adj_\tau, \mathbf{f}_\tau) = stateUpdate(Adj_{\tau-1}, Arcs, \mathbf{f}_{\tau-1})$;
6 $Q_\tau(s,a) = computeRL(Adj_\tau, R(\mathbf{f}_\tau), Q_{\tau-1}(s,a))$;
7 $Policy_\tau = \frac{e^{\frac{Q_\tau(s,a)}{T}}}{\sum_i e^{\frac{Q_\tau(s,a_i)}{T}}}$;

After that, the probabilistic distribution $P(G|O)$ of goals are computed according to Eq. (9). In line 4, $P(G|O)$ along with $Adj_{\tau-1}$ are fed into the DLNI module to generate the set of arcs $Arcs$ to be interdicted. Further, states would be updated based on $Arcs$, $Adj_{\tau-1}$ and $\mathbf{f}_{\tau-1}$. In lines 6 and 7, we compute the $Q$ value using reinforcement learning, and the $Policy$ under the maximum entropy assumption for the next stage.

## Experiments

We conducted extensive experiments on the basis of a human evader action data upon the Chicago Sketch Road Network (Lunday and Sherali 2010), which consists of 933 vertices and 2950 edges. We collect the moving trajectories from the same group of subjects in an interactive environment, as shown in Figure 3 (a), generated by the Unreal Engine 4 (Engine 2016). The human evader has one starting point and three predefined possible destinations which would be selected with equal probability at the beginning. Along with an IRL-learned model using human behavioral data, we give another two similar models predefined by human experts for comparison.

For the goal recognition part, we simplify the goal termination function as follows: If evader reaches its terminus, then the goal is achieved, otherwise it changes the goal with a probability of $0.01$ for every state. The observation, containing the evader's current position, of the recognizer would be missing with a probability of $0.2$. The computation of the SPNI is formulated into a BLMIP and solved using the MIP solvers of CPLEX 11.5 and YALMIP toolbox of MATLAB (Lofberg 2005). The $N_p$ of the particle system is set to 600.

## Tests on IRL-based Behavior Learning

The features are used to discriminate between locations in the environment. Based on the feature selected above, we compute the weight values for three predefined goals, as shown in Table 1, using the Maximum Entropy IRL in Algorithm 1.

Table 1: The weights of features for different goals using IRL

| Target No. | Distance $d$ | X | Y | Precision |
|---|---|---|---|---|
| 1 | 1.2 | 0.8 | 0.4 | 0.9290 |
| 2 | 1.5 | 0.9 | 0.5 | 0.8845 |
| 3 | 0.8867 | 0.8 | 1.1 | 0.8867 |

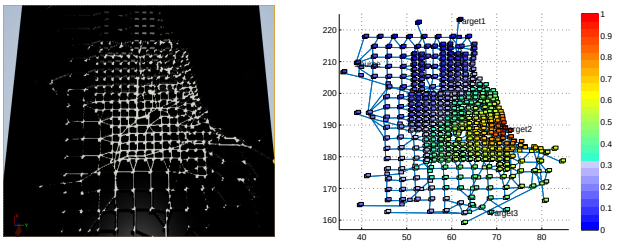
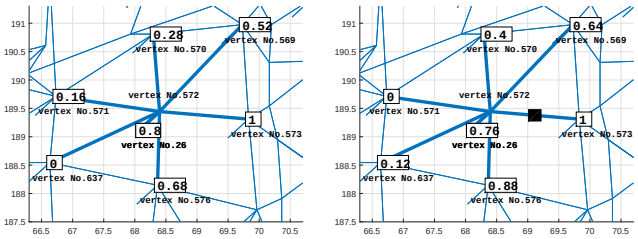



Figure 3: (a) The Chicago Sketch Roadmap for data collection and algorithm tests; (b)The final value map for **Target 2** computed by reinforcement learning using IRL-generated reward

Table 1 shows that the policy learned by IRL for **Target 1** is most closely related to the test trajectories, compared to **Target 2** and **3**. This is because there exists several non-distinctive paths for the latter two targets, which introduces ambiguity not only to human observer, but also to the IRL learner. Also, we show the final value map for **Target 2** computed by reinforcement learning using IRL-generated rewards, as shown in Figure 3 (b). These relative values are computed using Eq. (7). They increase from blue (value 0) to red (value 1), and show a general trend of the evader moving from the source point to **Target 2**.

Further, applying the behavior model to network interdiction, we can compute the change in the $Q$ value for each fixed vertex according to Eq. (8), after its neighboring edges are interdicted as shown in Figure 4. We select the index No.572 vertex at the position $(68, 189)$ to analyze, and know **Target 2** is located in the east of this mini-map in advance. The relative $Q$ values for No.572's neighboring vertices, which represent the set of evader's possible action selections and are connected using bold lines, are shown in the white boxes. Figure 4 (a) shows the original values before network interdiction. After the edge, labeled in the middle by a solid square as shown in Figure 4 (b) along the vertex No.572 and 573, is interdicted, relative $Q$ values change accordingly. Specifically, among 7 neighboring vertices, the relative $Q$ values increase at No.576, No.637, No.569 and No.570, and decrease at No.26 and No.571. Though the relative value for vertex No.573 is still the highest, its real value decreases after the interdiction, as the total increased $Q$ value of 4 vertices is higher than that of the decreased value. This is because the interdiction increases the cost for evader to traverse from No.572 to **Target 2** taking actions to No.573. This also explains the relative value changes for the other 6 vertices.




(a) relative $Q$ value before (b) relative $Q$ value after

Figure 4: Relative $Q$ value for action selections changes at vertex No.572 before and after network interdicted (Knowing the final goal **Target 2** locates in the east of this mini-map)

## Tests on Goal Recognition-assisted DLNI

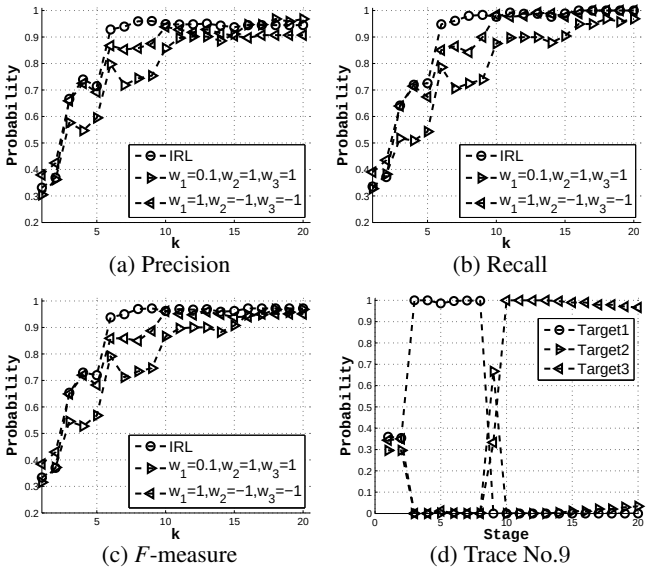
In this test, we first verify the effectiveness of IRL learned behavior model in goal recognition, using the dataset from the same group of subjects. Then we test the overall performance of Algorithm 2 in DLNI problem. The dataset consists of 100 labeled traces with each one possessing an average of 20.65 steps. There are approximately 19% traces where the evader's goal is changed at least once during half way. To show the details of goal recognition results using IRL learned behavior model, and especially in a situation where the goal has been changed midway, we select the first trace that has goal changes as an example, which is trace No.9, as shown in Table 2.

Table 2: The details of No.9 goal-changing trace

| Trace No. | Duration | Targets | Goal Interrupted |
|---|---|---|---|
| 9 | $\tau \in [1, 7]$ | Target 1 | Yes |
| | $\tau \in [8, 20]$ | Target 3 | No |

As shown in Table 2, the evader in trace No.9 selected **Target 1** to be its first terminus before changing it to **Target 3** at $\tau = 8$, and eventually reaching the goal at $\tau = 20$. Recognition results are shown in Figure 5 (d). The probability of the real goal **Target 1** increases quickly during the initial period. When the goal changes at $\tau = 8$, our method responds very quickly and the correct estimate is maintained until the end. Further, we compare the performance of goal

recognizers using different behavior models in Figure 5(a-c) by statistic metrics of precision, recall and $F$-measure, which are frequently used to measure overall accuracy of the recognizer (Sukthankar *et al.* 2014). To evaluate traces with different lengths, the paper applies the method in (Yue *et al.* 2016), and partitions the traces into $k$ stages. The weights of the other two cases are $\vec{\omega}_1 = (0.1, 1.0, 1.0)$ and $\vec{\omega}_2 = (1.0, -1.0, -1.0)$ respectively. The goal recognizer using IRL learned model has a faster reaction time and performs the best till the end in all three metrics.

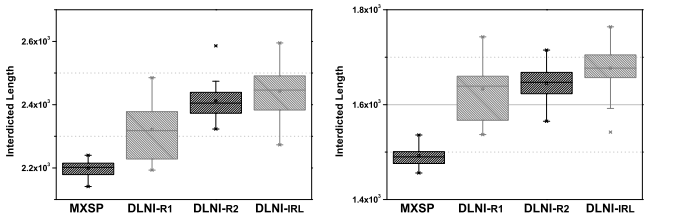



(a) Precision (b) Recall

(c) $F$-measure (d) Trace No.9

Figure 5: The comparison of goal recognizer using different behavior models and the performance of recognizing trace No.9 using IRL-assisted goal inference

Finally, we test the overall performance of Algorithm 2 in the DLNI problem, as shown in Figure 6, which compares the original static MXSP solver with DLNI solvers using different behavior models, i.e., IRL learned model (DLNI-IRL) and models (DLNI-R1 and DLNI-R2) with human specified $\vec{\omega}_1$ and $\vec{\omega}_2$. The DLNI solver using IRL learned behavior model performs best.




(a) Goal Unchanged ($G_1$) (b) Goal Changed Once ($G_1$ to $G_2$)

Figure 6: Comparison of shortest path maximum length using static MXSP solver and DLNI solvers with different behavior models ($G_1$=Target 2, $G_2$=Target 3, $t_{change}$=10).

## Conclusion

The paper proposes an *Inverse Reinforcement Learning* (IRL)-based opponent behavior modeling method, and successfully applies it in the goal recognition assisted Dynamic Local Network Interdiction (DLNI) problem. Experimental results indicate that our learned behavior model has a higher tracking accuracy and yields better interdiction outcomes than other models.

## Acknowledgments

The work is sponsored by the National Natural Science Foundation of China under Grants No.61473300.

## References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pages 1–8, 2007.

Dorit Avrahami-Zilberbrand and Gal A Kaminka. Fast and complete symbolic plan recognition. In *IJCAI*, pages 653–658, 2005.

Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

Francis Bisson, Hugo Larochelle, and Froduald Kabanza. Using a recursive neural network to learn an agent's decision model for plan recognition. In *IJCAI*, pages 918–924, 2015.

Hung Hai Bui, Svetha Venkatesh, and Geoff West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

Paola Cappanera and Maria Paola Scaparra. Optimal allocation of protective resources in shortest-path networks. *Transportation Science*, 45(1):64–80, 2011.

Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. User simulation in dialogue systems using inverse reinforcement learning. In *Interspeech 2011*, pages 1025–1028, 2011.

Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

Unreal Engine. Unreal engine 4. *Unity3D Engine*, 2016.

Christopher W Geib and Robert P Goldman. Plan recognition in intrusion detection systems. In *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, volume 1, pages 46–55. IEEE, 2001.

Andreas G Hofmann and Brian C Williams. Intent recognition for human-robot interaction. In *Interaction Challenges for Intelligent Assistants*, pages 60–61, 2007.

Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

Gilbert Laporte, Juan A Mesa, and Federico Perea. A game theoretic framework for the robust railway transit network design problem. *Transportation Research Part B: Methodological*, 44(4):447–459, 2010.

Lin Liao, Donald J Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007.

Diane J Litman and James F Allen. A plan recognition model for subdialogues in conversations. *Cognitive science*, 11(2):163–200, 1987.

Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2005.

Brian J Lunday and Hanif D Sherali. A dynamic network interdiction problem. *Informatica*, 21(4):553–574, 2010.

Wookhee Min, Eunyoung Ha, Jonathan P Rowe, Bradford W Mott, and James C Lester. Deep learning-based goal recognition in open-ended digital games. *AIIDE*, 14:3–7, 2014.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

David V Pynadath and Michael P Wellman. Generalized queries on probabilistic context-free grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):65–77, 1998.

Miquel Ramırez and Hector Geffner. Goal recognition over pomdps: Inferring the intention of a pomdp agent. In *IJCAI*, pages 2009–2014. IJCAI/AAAI, 2011.

Maria P Scaparra and Richard L Church. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923, 2008.

Marwaan Simaan and Jose B Cruz. On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555, 1973.

Gita Sukthankar, Christopher Geib, Hung Hai Bui, David Pynadath, and Robert P Goldman. *Plan, activity, and intent recognition: Theory and practice*. Newnes, 2014.

Bulent Tastan, Yuan Chang, and Gita Sukthankar. Learning to intercept opponents in first person shooter games. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 100–107. IEEE, 2012.

Kai Xu, Kaiming Xiao, Quanjun Yin, Yabing Zha, and Cheng Zhu. Bridging the gap between observation and decision making: Goal recognition and flexible resource allocation in dynamic network interdiction. In *IJCAI*, 2017.

Quanjun Yin, Shiguang Yue, Yabing Zha, and Peng Jiao. A semi-markov decision model for recognizing the destination of a maneuvering agent in real time strategy games. *Mathematical Problems in Engineering*, 2016, 2016.

Shiguang Yue, Kristina Yordanova, Frank Krüger, Thomas Kirste, and Yabing Zha. A decentralized partially observable decision model for recognizing the multiagent goal in simulation systems. *Discrete Dynamics in Nature and Society*, 2016, 2016.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.