entropy

MDPI

Article

Single Real Goal, Magnitude-Based Deceptive Path-Planning

Kai Xu, Yunxiu Zeng, Long Qin and Quanjun Yin *

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; xukai09@nudt.edu.cn (K.X.); zengyunxiu@nudt.edu.cn (Y.Z.); qinlong@nudt.edu.cn (L.Q.)
* Correspondence: yinquanjun@nudt.edu.cn; Tel.: +86-731-84576294

Received: 2 December 2019; Accepted: 6 January 2020; Published: 10 January 2020

Abstract: Deceptive path-planning is the task of finding a path so as to minimize the probability of an observer (or a defender) identifying the observed agent's final goal before the goal has been reached. It is one of the important approaches to solving real-world challenges, such as public security, strategic transportation, and logistics. Existing methods either cannot make full use of the entire environments' information, or lack enough flexibility for balancing the path's deceptivity and available moving resource. In this work, building on recent developments in probabilistic goal recognition, we formalized a single real goal magnitude-based deceptive path-planning problem followed by a mixed-integer programming based deceptive path maximization and generation method. The model helps to establish a computable foundation for any further imposition of different deception concepts or strategies, and broadens its applicability in many scenarios. Experimental results showed the effectiveness of our methods in deceptive path-planning compared to the existing one.

Keywords: deception; path-planning; plan recognition; information entropy

1. Introduction

Deceptive planning in an adversarial environment enables humans or AI agents to cover their real intentions or mislead an opponent's situation awareness. This would be of great help to many real-world applications, such as deceptive network intrusion [1], robotic soccer competition [2], intelligence reconnaissance [3], real-time strategy games, privacy protection [4], important convoy escorting, strategic transportation, or even military operations. Deceptive path-planning is one of its representative tasks. Masters and Sardina [5,6] first elaborated this problem and proposed three basic metrics, extent, density, and magnitude, to define deception. Among them, magnitude corresponds to any quantified measures of path deception at each individual node or step and has not been talked about yet.

This paper mainly uses the *magnitude* metric to measure the path deceptivity. To compute its value, based on the classical model-based goal recognition approaches [7], our method inverts recognition results and further uses them in planning the deceptive path. Goal recognition evaluates the posterior $P(G|O)$ over the possible goal set G given observation O; further, $P(G|O)$ is inversely used to quantify different forms of deception defined over each individual node. Further, we exploit the recent finding stated in [8], which has also been applied in [5], that goal recognition in path-planning does not need refer to any other historical observation given the observed agent's starting point and its current location. This finding enables that at any node, the posterior probability distribution upon all possible goals, along with the following deception values, could be precalculated at any time, and has nothing to do with the path taken to reach that node.

With the deception being defined upon each separate node or step, a deceptive path could be easily generated when we maximize deception along the path from the start to the end. Inspired by this intuition, the paper proposes a single real goal magnitude-based deceptive path-planning model, as well as a deceptive path generation model formed into a mixed-integer programming problem. The advantages of the model are generalized as follows: firstly, using magnitude to quantify deception at each node helps to establish a computable foundation for any further imposition of different deception strategies, and thus broadens the model's applicability in other scenarios with different deception characteristics; secondly, differently from the previous method [5] where deception could only be exploited from two goals (the real goal and a specific bogus goal), our method fully takes advantage over all possible goals during the generation of deceptive path; lastly, a moving resource could be flexibly defined to serve the deceiver's trade-off between deceptivity and resource.

The paper is organized as follows. Section 2 describes a single real goal magnitude-based deceptive path planning framework. Section 3 presents a single real goal magnitude-based deceptive path generation method formulated into a mixed-integer programming problem. We present several classical deceptive strategies which have been talked about under different names in many studies, along with a newly proposed one. Also, a cyclic deceptive path identification method is presented in order to handle cycles that may occur in the optimization results. We conclude with a case study on an 11×11 grid map, an empirical evaluation, and a conclusion.

2. Background and Related Work

This work lies at the intersection of two well-established and much-studied disciplines within computer Ssience: path-planning, which is the problem of finding a path through a graph or grid from a given start to a given goal [9–11]; and probabilistic goal recognition, which tries to infer an agent's goals that are evaluated continuously in the form of a probabilistic distribution over goals, given some or all of the agent's observed actions [7,12–14].

Grounded on the proposition that the order of goals ranked by likelihood can surely be inverted to describe their unlikelihood, the paper uses the results of the probabilistic goal recognition to plan a deceptive path using optimization method. Thus in this section, we first introduce the basic concepts, background, and applications of deception, especially the deception in path-planning. After that, we talk about the related works in goal or plan recognition, and review the recent developments of problem formulations, models, and algorithms applied in probabilistic goal recognition.

2.1. Deception and Deceptive Path-Planning

The deception problem is significant; it is common, and occurred frequently in human history [15]. As for its popularity, it is also a topic with a long history in computer science, particularly within the realms of artificial intelligence [16], non-cooperative game theory [3,17–19], and one of increasing relevance—in social robotics [20]. Deception is a key indicator for intelligence, as shown by a study investigating the role of working memory in verbal deception in children [21]. Intelligent agents, computer generated forces, or non-player characters who apply deceptive strategies are more realistic, challenging, and fun to play against [22] both in video games and serious training simulations. Furthermore, the potential use of deception has also been recognized in many multiagent scenarios, such as negotiation [23,24], multi-object auctioning [25], pursuit-evasion [26–28], and card games [29].

Defined by [30] and we quote here, deception is "the conscious, planned intrusion of an illusion seeking to alter a target's perception of reality, replacing objective reality with perceived reality". In a more dedicated definition [31], the deceptive tactics applied in the above applications could be further partitioned into two classes, denial (hiding key information) and deception (presenting misleading information). Tactics like masking, repackaging, dazzling, and red flagging are grouped in the denial type, while mimicking, inventing, decoying, and double play belong to the second one. These two patterns of deception with significant differences take turns at appearing in many studies, though the authors usually do not explicitly distinguish between them.

As a more focused area, research on the deceptive path appears in the literature under various guises. Jian et al. [32] tried to study the deception in path trajectories drawn by human subjects who had been asked beforehand to deceive an imaginary observer using a paper-and-pencil tests. Analyzed using both geographical and qualitative methods, the paper captured 38 recognizable characteristics, showing the existence of deception patterns and strategies in human behaviors, including *denial* and *deception*.

Hespanha et al. [17,18] and Root et al. [3] studied how deception could be used by rational players in the context of non-cooperative games. Hespanha [17] showed that, when one of the players can manipulate the information available to its opponents, deception can be used to increase the player's payoff. Interestingly however, when the degree of possible manipulation is too high, deception becomes useless against the intelligent opponent, as the opponent makes the decision as if there are no observations at all. This exactly accords with the objective of the *denial* strategy. Using the same strategy but in a more practical case, Root et al. [3] studied the deceptive path generation applied in UAVs' reconnaissance missions while under the opponent's surveillance and fire threat. In a domain modeled as a graph, the system selects a ground path, and then constructs a set of flight plans that involve overflying not only that path but every edge capable of supporting military traffic. The execution of the paths renders observation meaningless: the defender must select from multiple routes, all with the same probability.

Deception strategy (presenting misleading information) arose in a path-planning-related experiment carried out by roboticists Shim and Arkin [33], inspired by the food-hoarding behavior of squirrels. Computerized robotic squirrels visit food caches and, if they believe themselves to be under surveillance, also visit false caches (where there is no food). On the basis of observed activity, a competitor decides which caches to raid and steals whatever food she finds. In tests, the deceptive robots kept their food significantly longer than non-deceptive robots, confirming the effectiveness of the strategy.

Recent innovative work on goal recognition design (GRD) [34–38] could be seen as an inverse problem to deceptive path-planning. Standing on the side of the observer, the GRD problem tries to reduce goal uncertainty and advance the correct recognition through redesigning the domain layout. To do so, they introduce a concept named "worst case distinctiveness (wcd), measuring the maximal length of a prefix of a plan an agent may take within a domain before its real goal has been revealed. At first, the wcd is calculated and minimized relying on three simplifying assumptions [34], one of which assumes that the agents are fully optimal. Thus, the type of deception they design against takes more of a form like the denial strategy.

Interestingly, following the continued research on the GRD problem where the original assumptions are gradually relaxed, another form of deception appears in the literature as well. When considering suboptimal paths, the authors of [35] focused on a *bounded nonoptimal* setting, where an agent was assumed to have a specified budget for diverting from an optimal path. Additionally, as they presented, it suits for situations where deceptive agents aim at achieving time-sensitive goals, with some flexibility in their schedule. This exactly is the condition for the deception strategy (presenting misleading information) to be applied. Though holding a different perspective, the GRD problem provides valuable insights to the study of deceptive path-planning.

The most recent work by Masters et al. [5] presents a model of deceptive path-planning, and establishes a solid ground for its future research. In their work, three measures—magnitude (at each step), density (number of steps), and extent (distance travelled)—are proposed to quantify path deception. Focusing particularly on extent, they introduce the notion of last deceptive point (LDP) and a novel way of measuring its location. Also, the paper explicitly applies the "denial" and "deception" strategies, termed by Masters as "simulation" (showing the false) and "dissimulation" (hiding the real), in planning the deceptive path. Still, the work [5] has several shortcomings to overcome. Firstly, as discussed in the last section, the LDP concept is narrowly defined and would lower the deception performance in certain situations. Also, their model loses sight of people's needs in generating deceptive path with various path lengths, as resource constraints cannot be represented in their model. Lastly, though they have tried to enhance path deception by unifying "denial" and “deception”, e.g., additional refinements (π_3, π_4 as in [5]) under the original dissimulation strategy, their model lacks the ability to combine the two or even more in one framework.

A closely related topic to deceptive path-planning is the deceptive or adversarial task planning. Braynov [39] presents a conceptual framework of planning and plan recognition as a deterministic full-information simultaneous-moves game and argues that rational players would play the Nash equilibrium in the game. The goal of the actor is to traverse an attack graph from a source to one of few targets and the observer can remove one of the edges in the attack graph per move. The approach does not describe the deception during the task planning and provides no experimental validation. Taking a synthetic domain inspired by a network security problem, Lisý [40] defines the adversarial goal recognition problem as an imperfect-information extensive-form game between the observer and the observed agent. In their work, a Monte-Carlo sampling approach is proposed to approximate the optimal solution and could stop at any time in the game.

The research on deception arises in many other studies which we mention here only briefly: in cyber security [41], privacy protection [4,42], and goal/plan obfuscation [43,44].

2.2. Probabilistic Goal Recognition

The goal recognition problem has been formulated and addressed in many ways, as a graph covering problem upon a plan graph [45], a parsing problem over grammar [46–49], a deductive and probabilistic inference task over a static or dynamic Bayesian network [12,50–54], and an inverse planning problem over planning models [7,55–58].

Among the approaches viewing the goal or plan recognition as an uncertainty problem, two formulations appear and solve the problem from different perspectives. One focuses on constructing a suitable library of plans or policies [51,52], while the other one takes the domain theory as an input and use planning algorithms to generate problem solutions [7,55]. The uncertainty problem is first addressed by the work [51], which phrases the plan recognition problem as the inference problem in a Bayesian network representing the process of executing the actor’s plan. It was followed by more work considering dynamic models for performing plan recognition online [47,48,59,60]. While that offers a coherent way of modeling and dealing with various sources of uncertainty in the plan execution model, the computational complexity and scalability of inference is the main issue, especially for dynamic models. In [52], Bui et al. proposed a framework for online probabilistic plan recognition based on the abstract hidden Markov model (AHMM), which is a stochastic model for representing the execution of a hierarchy of contingent plans (termed policies). Scalability in policy recognition in the AHMM, and the inference upon its transformed dynamic Bayesian network, are achieved by using an approximate inference scheme known as the Rao–Blackwellised particle filter [61].

Within that formulation, research continues to improve the model’s expressiveness and computational attractiveness, as in [12,54,62–66]. Also, recently machine learning methods including reinforcement learning [66], deep learning [67,68], and inverse reinforcement learning [69,70], have already been successfully applied to learning the agents’ decision models for goal recognition tasks. These efforts once again extend the usability of policy-based probabilistic goal recognition methods by constructing agents’ behavior models from real data.

Recently, the work [55,56] shows that plan recognition can be formulated and solved using off-the-shelf planners, followed by work considering suboptimality [7] and partial observability [57]. Not working over libraries but over the domain theory where a set $\mathcal{G}$ of possible goals is given, this generative approach solves the probabilistic goal recognition efficiently, provided that the probability of a goal is defined in terms of the cost difference of achieving the goal under two conditions: complying with the observations and not complying with them. By comparing cost differences across goals, a probability distribution is generated that conforms to the intuition: the lower the cost difference, the higher the probability.

Several works followed that approach [58,71–74] by using various automated planning techniques to analyze and solve goal or plan recognition problems. Its basic ideas have also been applied in new problems, such as goal recognition design [34] and deceptive path-planning [5]. The advantages of the latter formulation are twofold: one is that by using plenty of off-the-shelf model-based planners the approach scales up well, handling domains with hundred of actions and fluents quite efficiently; the other one lies in the fact that the model-based method has no concerns about the recognition of joint plans for achieving goal conjunctions. Joint plans come naturally in the generative approach from conjunctive goals, but harder to handle in library-based methods.

According to the relationships between the observer and the observed agent, the goal or plan recognition could be further divided into keyhole, intended, and adversarial types [1,75–77]. In keyhole plan recognition, the observed agent is indifferent to the fact that its plans are being observed and interpreted. The presence of a recognizer who is watching the activity of the planning agent does not affect the way he plans and acts [59,78]. Intended recognition arises, for example, in cooperative problem-solving and in understanding indirect speech acts. In these cases, recognizing the intentions of the agent allows us to provide assistance or respond appropriately [75]. In adversarial recognition, the observed agent is hostile to the observation of his actions and attempts to thwart the recognition [79,80].

3. The Single, Real Goal, Magnitude-based Deceptive Path-Planning

We first begin by formalizing path planning and goal recognition. We propose a rather simple but general model suitable for different deceptive path planning domains. We assume that there are two players: one is the evader (the observed agent); the other one is the defender (the observer). The evader chooses one goal from a set of possible goals and attempts to reach it, whereas the defender must correctly recognize which goal has been chosen. We assume that the outcomes of the action are deterministic and the model is fully observable to the evader and the defender, meaning that the environment and each others' actions are visible to both players. In addition, the evader has several assumptions about the defender. Firstly, the defender does not know the decision-making process of the evader. Secondly, the defender operates under a fixed recognition algorithm.

The notion of a road network captures the intuition about all possible deceptive paths that can be planned against a set of goals.

Definition 1. *A road network is a triplet $RN = \langle N, E, c \rangle$, where:*

- *N is a non-empty set of nodes (or locations);*
- *$E \subseteq N \times N$ is a set of edges between nodes;*
- *$c : E \to \mathbb{R}_0^+$ returns the length of each edge.*

Define the path π as a sequence of nodes through a road network, and π^i as the i-th node in π. The cost of path π is the sum of length of all the edges in π, and we have $cost(\pi) = \sum_{i=0}^{k-1} c(\pi^i, \pi^{i+1})$. In its formal definition, the path-planning problem considers a set of goals; in our setting, however, we give a single-goal path-planning problem as follows.

Definition 2. *The single-goal path-planning problem is a triplet $SGPP = \langle RN, s, g \rangle$, where:*

- *$RN = \langle N, E, c \rangle$ is the road network;*
- *$s \in N$ is the source;*
- *$g \subseteq N$ is the single goal/destination/target.*

The solution to the single-goal path-planning problem is a path which starts in its source s and ends in one single goal g. Denote $optc(n_i, n_j)$ as the cost of the optimal solution from the node n_i to n_j.

Definition 3. *The probabilistic goal recognition problem is a tuple $PGR = \langle RN, s, G, O, Prob \rangle$, where:*

- *$RN = \langle N, E, c \rangle$ is the road network;*

- *$s \in N$ is the source;*
- *$G = \{g_r\} \cup G_b$ is the possible goal set, where g_r is the single real goal, and G_b is the set of bogus goals;*
- *$O = o_1, \ldots, o_{|O|} \in N$ is the observation sequence;*
- *Prob are the prior probabilities upon the possible goal set G.*

A solution to the probabilistic goal recognition is a conditional probability distribution $P(G|O)$ across G given observation sequence O, with each two neighboring observations $o_i, o_{i+1} \in O$ not needing to be temporally adjacent. The quality of solution, or we could say that of the goal recognition algorithm, could be reflected from whether it has $P(g_r|O) \geq P(g|O)$ for all $g \in G_b$. Further, in many real-world applications with high risk and interest, only $P(g_r|O) \geq P(g|O)$ in PGR's solutions may not be enough.

In this paper, we take a concept *costdif* which was applied early in [7] to reflect the goals' uncertainties at each node or location. Specifically, it captures the intuition that *costdif* has an inverse correlation to goals' posterior probabilities; that is

Theorem 1. *$P(g|O) > P(g'|O)$ if and only if costdif$(s, g, O) <$ costdif(s, g', O).*

The *costdif*(s, g, O) first defined in [7] represents the cost difference between the optimal cost $optc(s, g, O)$ of plans that accords with observation and $optc(s, g, \bar{O})$ that is not. The posterior probability distribution upon goal set G could be computed using:

$$P(G|O) = \alpha \frac{e^{-\beta X}}{1 + e^{-\beta X}} \tag{1}$$

where $X = costdif(G, O)$, α is a normalizing constant across goals and β is a positive constant which captures a "soft rationality" assumption.

As we have talked about in the last section, Masters and Sardina [8] proved that in path-planning domain, goal recognition needs not refer to any other historical observation given the observed agent's source and current location. In other words, only the final observation in the sequence that matters:

$$costdif_{MS}(s, g, n) = optc(n, g) - optc(s, g) \tag{2}$$

where $n = O^{|O|}$ is the last observed location of the evader. Like in Theorem 1, we have:

Theorem 2. *$P(g|n) > P(g'|n)$ if and only if costdif$(s, g, n) <$ costdif(s, g', n).*

Both Theorem 2 and Equation (2) enable that at any node, we could precalculate the posterior probability distribution upon all possible goals in an offline manner.

Definition 4. *The single real goal deceptive path-planning problem is a tuple $SRGDPP = \langle RN, s, G, g_r, P \rangle$, where:*

- *$RN = \langle N, E, c \rangle$ is the road network;*
- *$s \in N$ is the source;*
- *$G = \{g_r\} \cup G_b$ is the possible goal set;*
- *g_r is the single real goal;*
- *$P(G|O)$ denotes the posterior probability distribution upon G given O.*

Specifically, in this paper we could replace $P(G|O)$ as $P(G|n)$ where n is the last node in that sequence. P stands for the observation model of the defender. We assume that G consists of all the possible goals (e.g., multiple highway entrances, airports, or harbors in a city), including a single real goal g_r and other bogus goals $g_b \in G_b$. The evader starts from s and moves towards g_r, while during the process, making use of both the possible goal sets G as well as the defender's observation model P to plan a deceptive path. Corresponding to the single-goal path-planning problem, this work considers a single real goal situation and leaves the multiple-real-goals setting to the future research. We assume that the evader has different deception strategies on planning the path.

Definition 5. *The evader's deceptive strategy is a function $F : P(G|i) \rightarrow mag_i$, where $P(G|i)$ is the set of probability distributions at the node $i \in N$ of road network RN over the possible goals G, mag_i is the deceptive magnitude at node i.*

Different strategies evaluate a different deceptive magnitude at each node. Later, we introduce strategies in detail. Further, we present a single, real goal, magnitude-based deceptive path planning model as follows:

Definition 6. *The single real goal, magnitude-based deceptive path-planning is a triplet $SRGMDPP = \langle SRGDPP, F, R \rangle$, where:*

- *$SRGDPP = \langle RN, s, G, g_r, P \rangle$ is the single real goal deceptive path-planning problem;*
- *$F : P(G|i) \rightarrow mag_i$ returns the deception magnitude value assigned to each node;*
- *R is the total amount of distance allowed for the deceiver traversing $s - g_r$ path.*

Apart from the elements in SRGDPP problem, the SRGMDPP adds a deceptive strategy function that maps the probabilistic goal recognition results into quantified path deception at each individual node. Also, a explicit presentation of total resource constraint enables the evader to flexibly adjust its deceptive path plan according to the time (or fuel) constraints in practice. We say that SRGMDPP provides a computable foundation for further impositions of different deception concepts or strategies, and broadens the model's applicability in other scenarios.

4. The Single Real Goal Magnitude-Based Deception Maximization

Inspired by the intuition that a deceptive path could be generated when maximizing deception along the s–g_r path, and following the above definitions, we present its mathematical formulation as follows.

4.1. Model Formulation

Denote $i, j \in N$ as the nodes and $k = (i, j) \in E$ the edge in the road network. The edge sets $FS(i)$ and $RS(i)$ represent the set of edges directed out of and into the node i. Assign the same magnitude value mag_i of node i to values mag'_k of all the edges k in the set $RS(i)$, where mag'_k is the deceptive magnitude associated with edges. The formulation is:

$$\textbf{[SRGMDM-P]} \quad \max_{\mathbf{x} \in X} \sum_{k \in E} (1 + mag'_k) \cdot c_k \cdot x_k$$

$$\sum_{k \in FS(i)} x_k - \sum_{k \in RS(i)} x_k = \begin{cases} 1 & \text{for} \quad i = s \\ 0 & \forall i \in N \backslash \{s, g_r\} \\ -1 & \text{for} \quad i = g_r \end{cases} \tag{3}$$

$$x_k \geq 0, \quad \forall k \in E \tag{4}$$

where c_k (the vector form $\mathbf{c}$) is the cost of traversing edge k; x_k (the vector form $\mathbf{x}$) is the integer variable controlling whether the evader traverses the edge k or not, $x_k = 1$ if the edge k is traversed, or else, $x_k = 0$. The solution $X = \{\mathbf{x} \in \{0,1\}^{|E|} | \mathbf{c}^T \mathbf{x} \leq R\}$. Equation (3) is the flow-balance constraint, which guides the evader to leave s and reach g_r and guarantees the nodes i in the set $N \backslash \{s, g_r\}$ to be visited and leaved at the same number of times. Soon, we give three kinds of measures of deceptivity: two were first talked about in [5]; one is proposed based on them.

4.2. Deceptive Strategy

Definition 7. *The simulation strategy* $F_{sim}(P(G|i)) = \max_{g_b \in G \setminus \{g_r\}} P(g_b|i) - P(g_r|i)$.

The simulation strategy measures the amount by which a false goal dominates the real goal. And the greatest deception occurs when there are the greatest dominance. F_{sim} drives the defender believes that the evader's target is one bogus goal ("showing the false"), and thus deploys resource to the wrong location.

Definition 8. *The dissimulation strategy* $F_{dissim}(P(G|i)) = \sum_{g \in G} P(g|n) \times \log_2(P(g|n))$.

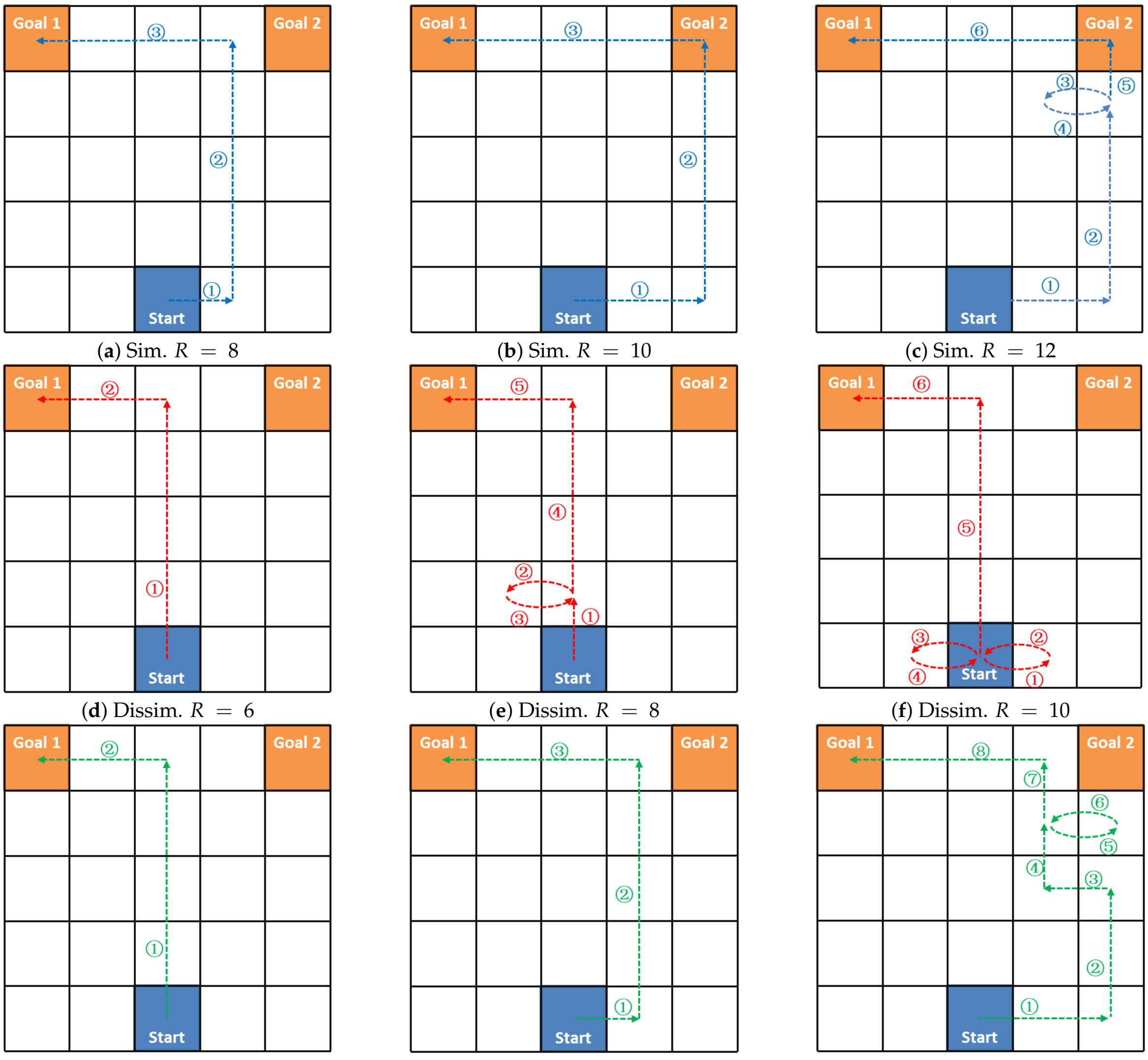
As a popular method, this type of deception (or we say the degree of ambiguity) is defined using the Shannon's entropy, which is usually used to describe information uncertainty. By taking advantage of the defender's observation model, the F_{dissim} strategy plans a path where different goals share similar posterior ("hiding the real"), and thus making the defender confused about where to deploy its resource. Instead, the defender will make decisions, as there are no observations at all, as discussed about in [17], though it may also guess correctly. Also, this strategy could prove its effectiveness in applications with high risk or interest, as without certainty it is itself a danger. Usually, compared to a simulation, this strategy consumes less resource.

However, both concepts of deception reflect only a fraction of what it means to be deceptive. The difference lies in the extent to which people want to deceive the observer. According to the strength of intention protection desire, usually these two strategies could be balanced in human behaviors. Thus the paper proposes a more flexible weighted combination strategy.

Definition 9. *The combination strategy* $F_{combi}(P(G|i)) = \omega_1 \cdot F_{sim} + \omega_2 \cdot F_{dissim}$, *where* $\omega_1 + \omega_2 = 1$.

We quantify that aspect of deception as a linear combination of the simulation and dissimulation. If we deceive successfully using the combination strategy, the defender would either be led to the false goal (when ω_1 is larger), or choose randomly in a set of possible goals (when ω_2 is larger).

Take the example, as shown in Figure 1, where the paper presents groups of deceptive paths optimized using SRGMDM-P under different resource constraints. Specifically, Figure 1a–c follows simulation (Sim.) strategy, Figure 1d–f accords with dissimulation (Dissim.), while paths in Figure 1g–i respect the combined one (Comb.). Several trends need to be noted here. When resources are insufficient for the agent planning the most deceptive way, the model returns a less deceptive version; e.g., the path in Figure 1a with $R = 8$ is less deceptive than the ones in Figure 1b or Figure 1c. In this case, it is the same as the one in Figure 1h following the combined strategy. A cyclic path appears when the deceiver is given additional resource, as we can see in Figure 1c,e,f,i. This satisfies the needs of those agents who either are time-sensitive to other cooperative missions or just have abundant resources. Paths generated following the combined strategy would be more like those under dissimulation when resources are inadequate (e.g., Figure 1d,g, but transform more like the ones under simulations with more resources coming in (refer to Figure 1c,i).



(**a**) Sim. $R = 8$ (**b**) Sim. $R = 10$ (**c**) Sim. $R = 12$

(**d**) Dissim. $R = 6$ (**e**) Dissim. $R = 8$ (**f**) Dissim. $R = 10$

(**g**) Comb. $R = 6$ (**h**) Comb. $R = 8$ (**i**) Comb. $R = 12$

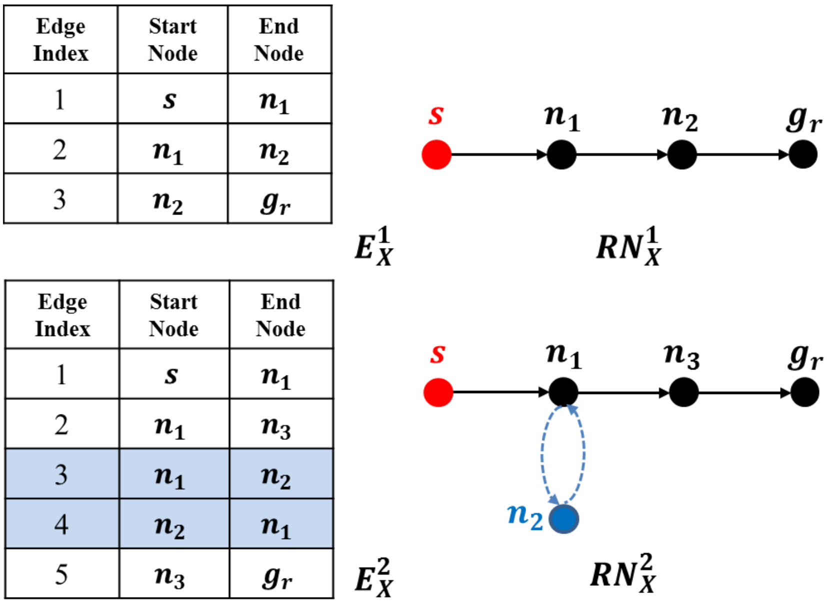
Figure 1. The deceptive path generated using the MDM model under different resource constraint R.

4.3. Cyclic Deceptive Path Identification

The result of the SRGMDM model is the optimized variable X, whose nonzero elements comprise the set E_X of edges that would be traversed by the evader. From the edge set E_X, a solution road network $RN_X = \langle N_X, E_X, c_X \rangle$ could be further generated, where N_X is a non-empty set of nodes that appears in the edges from E_X, and c_X denotes the length of original edge.

When no circles exist in the deceptive path, for example, in the optimal path planning problem, given start position s and destination g_r, the agent's moving trajectories could be easily sketched out by linking edges from end to end. For example, in the first case shown in Figure 2, where $E_X = \{(s, n_1), (n_1, n_2), (n_2, g_r)\}$, the corresponding path is $s \to n_1 \to n_2 \to g_r$. However, when cyclic paths occur, as in the case below, where $E_X = \{(s, n_1), (n_1, n_3), (n_1, n_2), (n_2, n_1), (n_3, g_r)\}$, this simple method may fail as the edges 3 and 4 (edges (n_1, n_2) and (n_2, n_1) marked in blue) will not be traversed by the evader.

An intuitive way to generate cyclic path is to try exhaust every possible combination that could cover all edges in E_X. For example, we could change the orders of edge numbers 2 and 3 in the above case, and get the path $s \to n_1 \to n_2 \to n_1 \to n_3 \to g_r$. However, it wiould become more complicated when much more circles exist in one path, as is the case in Figure 1f.



Edge Index	Start Node	End Node
1	s	n_1
2	n_1	n_2
3	n_2	g_r

Edge Index	Start Node	End Node
1	s	n_1
2	n_1	n_3
3	n_1	n_2
4	n_2	n_1
5	n_3	g_r

Figure 2. The deceptive paths with and without cycles on two solution road networks, RN_X^1 and RN_X^2.

Before introducing our method for identifying the cyclic deceptive path from E_X, the paper first considers the situation when cyclic path could appear in a solution road network.

Definition 10. *In a solution road network $RN_X = \langle N_X, E_X, c_X \rangle$, the cyclic path appears when it satisfies either one of the following conditions:*

- $|FS(s)| - |RS(s)| = 1$*, and* $|FS(s)| > 1$*;*
- $\exists i \in N_X \backslash \{s, g_r\}, |FS(i)| = |RS(i)| > 1$.

The nodes following the above conditions are grouped by the set N'_X.

The edge sets $FS(i)$ and $RS(i)$ represent the sets of edges directed out of and into the node i. The nodes in N'_X locate on cyclic trajectories that may appear in the whole path.

Theorem 3. *In the solution road network $RN_X = \langle N_X, E_X, c_X \rangle$, consider the case where $|FS(i)| = 2$ for $i \in N_X$; there will always exist one edge $j \in FS(i)$ whose ending node has a way of returning back to i.*

Proof. Assume to the contrary that all the ending nodes of edges in $FS(i)$ have no way returning back to i; then, according to the constraints in Equation (3), there must be edges starting from the other node (denoted as j) that direct to node i. This in turn means that node j cannot be reached from i and j must be in earlier positions than i of the whole path. Then, consider two cases: (1) when $i = s$, node i itself locates in the first place before any other nodes; (2) for $i \in N'_X \backslash \{s\}$, when the node i is reached, a trajectory starting from s must have been drawn out already. Then, j is either on this trajectory, or it is not. When j is on, it cannot contribute one more edge into i. When it is not, two trajectories, $(s, \ldots, j, i)$ and the other one $(s, \ldots, i)$ with no j, coexist, and that contradicts the task of generating only one path from E_X. □

The above theorem could be easily extend to more general situations.

Theorem 4. *For the node $i \in N_X$, there will always exist $|FS(i)| - 1$ numbers of distinct paths returning back to i from the ending nodes of edges $j \in FS(i)$.*

The above observations assure us that when confronted with nodes $i \in N'_X$, we could first find passable ways starting from $FS(i)$ to i, add these edges into the ordered edge set, and continuously do these two steps until no elements in $FS(i)$ could return back to i. Relying on Theorems 3 and 4, we present the *cyclic-path-identify* algorithm (whose pseudocode is given in Algorithm 1).

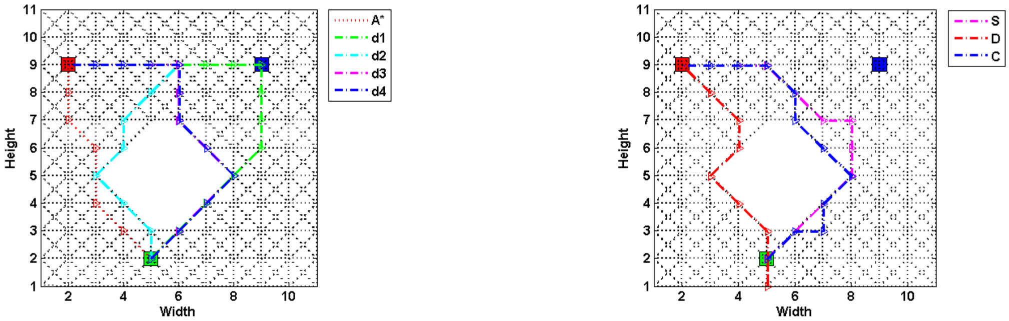
Algorithm 1: *Cyclic-path-identify algorithm*

```
   Input: Original graph $RN = (N, E, c)$, optimized result $X$, source $s$, real goal $g_r$
   Output: Ordered edge set $Path$
1  Retrieve the solution road network $RN_X = (N_X, E_X, c_X)$ from $RN$ and $X$;
2  Initialize $Path = \{s\}, temp_n = s, temp_E = E_X$;
3  while $temp_E \neq null$ do
4      if ($temp_n \neq g_r$ and $|FS(temp_n)| > 1$) then
5          foreach edge $e \in FS(temp_n)$ do
6              $j \leftarrow$ the end node of edge $e$;
7              $(traj, flag) = isPassable(j, temp_n, RN_X)$;
8              if $flag$ then
9                  $Path = Path \cup \{nodes\ in\ traj\}$;
10                 $coverEdgeSet = getEdgeFromPath(traj)$;
11                 $temp_E(coverEdgeSet) = null$;
12                 $temp_n = traj(end)$;
13             else
14                 continue;
15             end
16         end
17     else
18         $j \leftarrow$ the end node of edge $FS(temp_n)$;
19         $Path = Path \cup \{j\}$;
20         $e \leftarrow$ the $(temp_n, j)$ edge;
21         $temp_E(e) = null$;
22         $temp_n = j$;
23     end
24 end
```

Algorithm 1 shows how to identify the cyclic path from the optimization result X, given the original road network $RN = (N, E, c)$ and the source s. We first retrieve the solution road network $RN_X = (N_X, E_X, c_X)$ from RN_X and X, and initialize an ordered edge set *Path* (Lines 1–2). Only when all edges in E_X have been added into *Path*, the process (Lines 4–23) can stop. For nodes not located on cyclic trajectory (Situation 1), they could be simply added to *Path* using an end-to-end style (Lines 18–22). While for those on the cycles (Situation 2), judged by conditions we have presented above, the method generates all the trajectories *traj* starting from end nodes j of $FS(temp_n)$, and adds the covered edges *coverEdgeSet* in trajectories into *Path*. This process (Lines 4–16) relies on Theorem 3 and 4, as they two guarantee that when facing cycles, there will always exist $|FS(i)| - 1$ numbers of distinct paths returning back to i from the ending nodes of edges $j \in FS(i)$. Through linking these distinct paths to *Path* in $|FS(i)| - 1$ times, finally the problem degrades from Situation 2 to Situation 1. Two additional steps are required for the process under both conditions: (1) deleting the covered edges from $temp_E$; (2) directing $temp_n$ to the new node. The A* path-finding algorithm is used in the *isPassable* function at line 7, where *traj* is a generated path starting from the node j and ending at $temp_n$, *flag* signals whether a path could be found.

5. Experiments

In this section, we compare the performance of the model and strategies proposed by this paper with those (strategies *d1* to *d4*) by [5]. First we present a case study upon a fixed 11 × 11 grid map to elaborate our method's performance, as shown in Figure 3. Methods are compared by metrics including generation time, path costs and their deceptivity, as shown in Table 1. Also, a probabilistic goal recognition method applied in [54] generates a posterior probability distribution $P(G|O)$ across possible goal set G given O, as shown in Figure 4. We set the weights of *combination* strategy to $\omega_1 = \omega_1 = 0.5$ to have a balanced performance between "showing the false" and "hiding the real." The computation of the SRGMDM was formulated into a mixed-integer programming and solved using the solvers of CPLEX 11.5 and YALMIP toolbox of MATLAB 2012 [81].



(**a**) Strategies in [5]

(**b**) Strategies in this paper

Figure 3. The deceptive path following the four strategies proposed in [5], and three in this paper upon a simple 11 × 11 grid-based road network, where the green square is the source, red is the single real goal g_r, and blue is the bogus one. An optimal path generated by A* is given as a comparison. An underlying assumption is that the agent would traverse the road network at a pace of one grid a time step.

Specifically, in Figure 3a, we show four strategies in [5] along with an optimal path-planning case (using A* algorithm) as a comparison. Strategy *d1* (the green dashed line), which is treated as a simplest "showing the false" strategy in [35], first takes an optimal path towards a bogus goal (marked as a blue square). Computationally inexpensive, this achieves a strongly deceptive path (referring to Table 1). However, the cost of its path is usually very high and, although it initially deceives both human and automated observers, reaching but not stopping at the bogus goal (which is the blue square in this case) immediately signals to a human that it is not the real goal. This is also evident as in Figure 4b, where $P(g_r|O)$ immediately rises high when the evader moves away from the bogus goal after time Step 8.

Path *d2* (the magenta dashed line) is regarded as a simplest "hiding the truth," by which the evader takes an optimal path π_{d2} directly from the source s to the last deceptive point (LDP), and then on to g_r. As we have talked about in the related works, the LDP is a concept and a node defined in [5], in which the authors believe beyond this point all nodes in the path are truthful (a truthful node is one at which $P(g_r|O) > P(g_b|O)$ for all $g_b \in G_b$). In Figure 4, we mark the part of path behind the LDP as red)

It might be more deceptive to a human observer until later in the path than π_{d1} above. However, π_{d2} is only weakly deceptive; that is, truthful steps are likely to occur before the LDP without additional checks and balances. As we could find in Table 1, the goal recognition cannot really tell which one is the right one. While in Figure 4c, this trend is clearer than the posterior probability of the real goal, that of the bogus goal, alternatively, becomes the larger one.

Strategies *d3* and *d4* are two refinements of *d2*. A path following the strategy *d3* can be assembled using a modified heuristic so that, while still targeting the LDP, whenever there is a choice of routes, it favors the bogus goal, increasing its likelihood of remaining deceptive. *d4* prunes all truthful nodes during the search from source s to the LDP, thereby requiring more time compared to the other three. However, upon this simple case study, there does not exist too much difference between *d3* and *d4*.

Table 1. The details of traces generated using different strategies, and the deceptivity (**1** = deceptive, **0** = correct prediction) of path when tested at 10%, 25%, etc., of their path lengths *prior to* the last deceptive point (LDP). (**St** = strategy, **C** = path cost, **T** = generation time ($\times 10^{-3}$s))

St	**C**	**T**	**10%**	**25%**	**50%**	**75%**	**99%**
A*	8.24	0.69	1	1	0	0	0
d1	15.66	2.37	1	1	1	1	1
d2	13.07	4.93	1	0	0	0	1
d3	13.07	3.81	1	1	1	1	1
d4	13.07	66.08	1	1	1	1	1
S	13.07	849.81	**1**	**1**	**1**	**1**	**1**
D	11.07	220.79	1	1	1	0	0
C	13.07	145.23	**1**	**1**	**1**	**1**	**1**

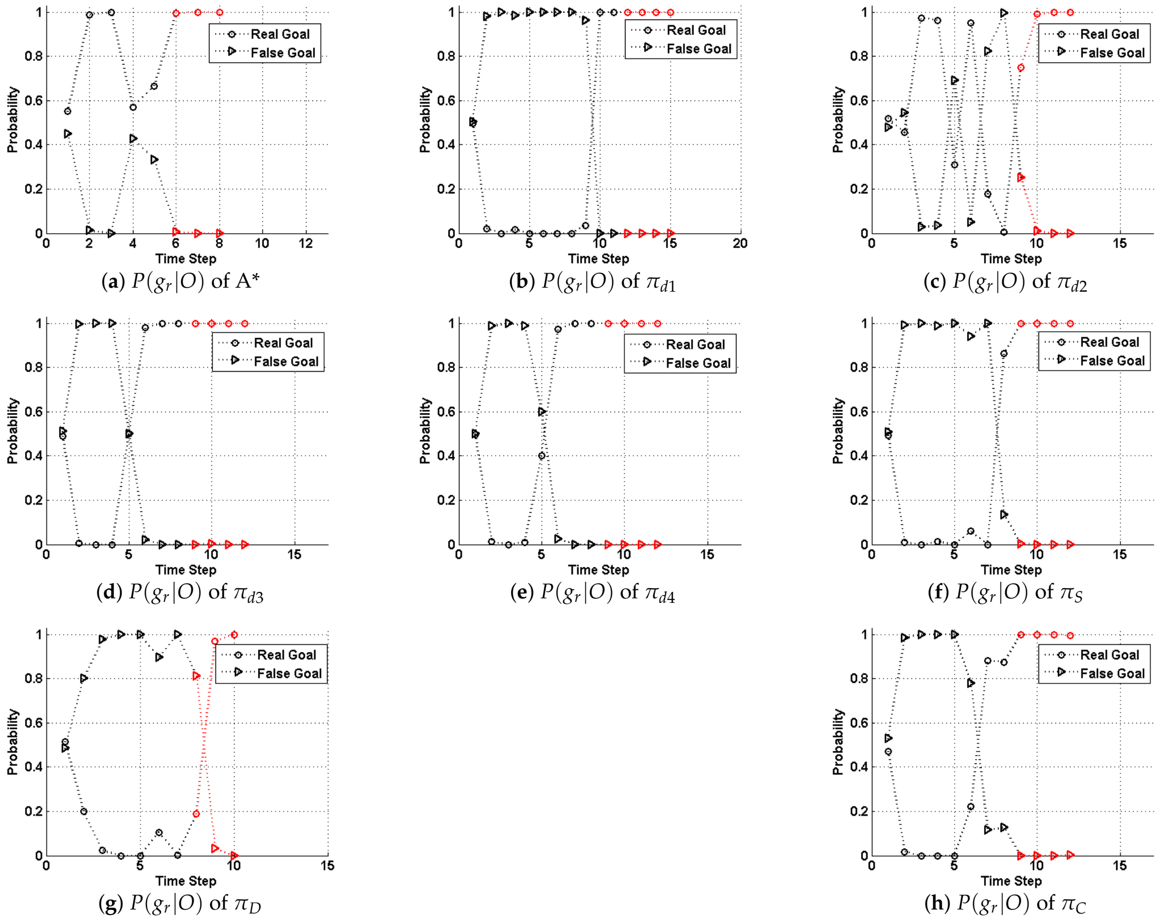
In Figure 3b, we show the paths generated using the SRGMDM model, where the deceptivity is measured by its magnitude at each step, following the simulation "S," dissimulation "D," and combination "C" strategies defined in this paper. Currently, we constrain the path cost of all three to no more than 13.07, the same as *d2*, *d3* and *d4*, but smaller than *d1*. From the results shown in Table 1, paths following *S* and *C* fully deceive the defender at each 10%, 25%, etc., of their path length prior to the LDP, while *D* strategy generates a cyclic path at the beginning (marked as red dashed line in Figure 3b) and loses its deceptivity in the last two-fifths of path length. This is reasonable, as both *d2* and *D* only try to make the defender not know the real goal, though it may still guess correctly. Compared to strongly-deceptive strategies *d1* and *S*, "showing the false" is not their primary focus, and thus in most situations they are more efficient in terms of cost.

As for the results of strategies *S*, *D*, and *C* shown in Figure 3, we could see that, before the LDP, all strategies deceive the defender well. For example in Figure 4f, the $P(g_r|O)$ is continuously dominated by that of the bogus goal until time step 7. While after the LDP, the goal recognizer successfully identifies the real goal behind the evader's misleading actions. It also should be noted that, the goal recognition method we use here is more sensitive to observations, as it captures the evader's real goal several steps ahead of the LDP in cases like Figure 4b,d,e,h.

The case study shows the effectiveness of our method in planning deceptive paths. Also, paths generated using SRGMDM do not rely on the so-called LDP as a transit point before moving towards the real goal, and the trajectories are more flexible, as shown in Figure 3. As LDP is calculated only according to the real goal g_r and the bogus goal which is the closest to g_r, as stated in [5]; our method instead fully takes advantage of the global information (including all possible goals) during the deceptive path-planning. Besides, the method's flexibility also shows, as cyclic deceptive paths would be generated whenever the moving resource is abundant.

In order to have a more thorough comparison between different methods and strategies, in the following experiments, we conduct an empirical evaluation over a problem set which is generated upon the grid-based maps from the moving-AI benchmarks [82]. The source and the real goal are already available in the benchmark. Bogus goals are added randomly at different locations. Figure 5 shows four 2D grid map examples that are selected for tests from the benchmarks.

As with the above cast study, in each case we generated one optimal path using the standard A* and seven deceptive paths, each corresponding to a different deceptive strategy. Using probabilistic goal recognition and under the assumption that priors for all goals are equal, we calculated probabilities at intervals to confirm/assess deceptive magnitude. As to evaluating traces with different lengths, the paper normalizes traces with different lengths into t $(t = 1, 2, ..., N_{stage})$ intervals. N_{stage} is the total number of intervals that we want to partition.



(**a**) $P(g_r|O)$ of A* (**b**) $P(g_r|O)$ of π_{d1} (**c**) $P(g_r|O)$ of π_{d2}

(**d**) $P(g_r|O)$ of π_{d3} (**e**) $P(g_r|O)$ of π_{d4} (**f**) $P(g_r|O)$ of π_S

(**g**) $P(g_r|O)$ of π_D (**h**) $P(g_r|O)$ of π_C

Figure 4. Probabilistic goal recognition results of the paths planned following the strategies shown in Figure 3. The section of $P(g_r|O)$ whose path after the last deceptive point is marked red.

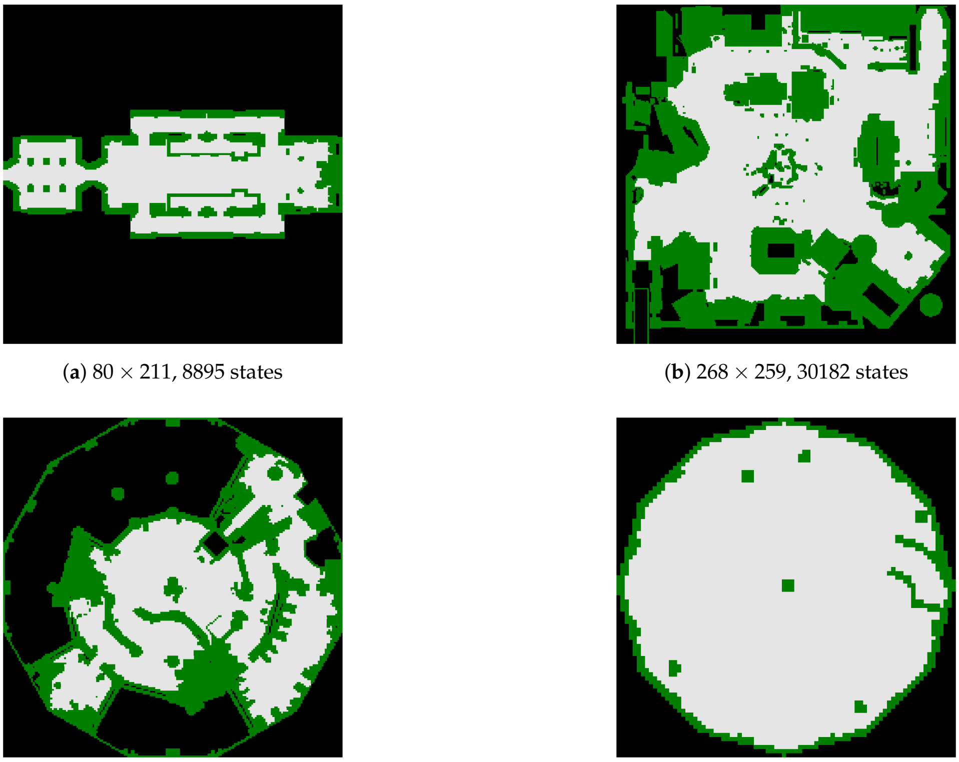
We got the statistical results using the metric of *F-measure*, which is frequently used to measure the overall accuracy of the recognizer [14] and is computed as:

$$precision = \frac{1}{|G|}\sum_{m=1}^{|G|}\frac{TP_m}{TI_m}, recall = \frac{1}{|G|}\sum_{m=1}^{|G|}\frac{TP_m}{TT_m} \tag{5}$$

$$F\text{-}measure = \frac{2 \cdot precision \cdot recall}{precision{+}recall} \tag{6}$$

where $|G|$ is the number of possible goals; TP_m, TI_m, and TT_m are the true positives, total of true labels, and the total of inferred labels for class $g_m \in G$ respectively. The value of *F-measure* will be between 0 and 1, and a higher value means a better confidence of recognizing the right goal while a less deceptive strategy. Thus, the lower value, the more deceptive.

The results are shown in Figure 6. The weights of *combination* strategy were set the same as in the above case study. The resource constraints of paths following simulation, dissimulation, and combination were the same as those of paths using *d2* strategy. As a general trend, we could find that the path deceptivity gradually decreased as more observation coming in. A* provides a performance baseline for comparisons with other strategies. Among all the strategies, the *F-measure* of A* increases greater than 80% early from interval $t = 2$, indicating the disappearance of path deceptivity. As stated in their work [5], strategy *d2* and *d3* are weakly deceptive, while *d1* and *d4* are strongly deceptive. Similarly, in our results, apart from A*, *d2* and *d3* generate the least deceptive paths in most situations. As for *d1*, different from their findings however, it performs the best in the first half (interval $t \leq 2$) of deceptive path. However, as its paths do not stop at the closest bogus goal, the goal recognizer immediately changes its belief to the real goal; thus, when $t \geq 4$, the performance of *d1* deteriorates the fastest among all the others. This is because the goal recognition algorithm we use considers more history information during the whole process while the method in [5] is not. From the results, we admit that *d4* is the most deceptive strategy compared to *d1*, *d2*, and *d3*.



(**a**) 80 × 211, 8895 states

(**b**) 268 × 259, 30182 states

(**c**) 194 × 194, 10557 states

(**d**) 84 × 84, 4706 states

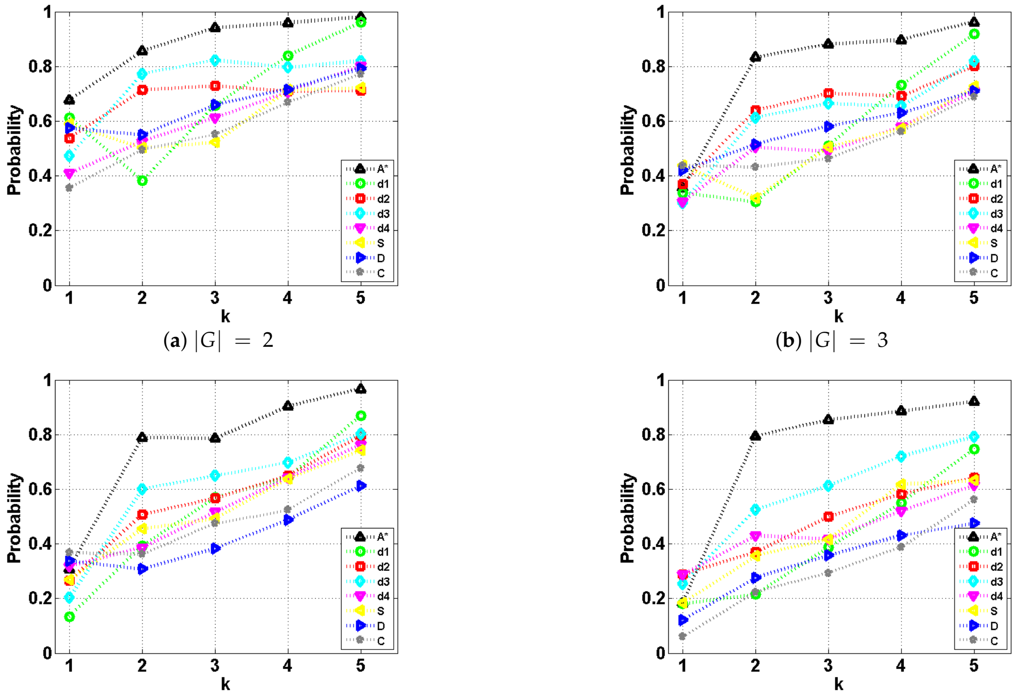
Figure 5. Four large-scale 2D grid maps from the moving-AI benchmarks.

Now, we compare the performance of *d4* with that of *simulation*, *dissimulation* and *combination*. Firstly, we could find that in all situations, the *simulation* strategy always has a similar (when $|G| = 3, 4, 5$) or even better performance (when $|G| = 2$) compared to *d4*.

Interestingly, for *dissimulation* and *combination* strategies, paths become more deceptive as the number of possible goals $|G|$ increases. Specifically, *dissimulation* is better than *d4* when $|G| > 3$. It continuously drags the value of *F-measure* down from 0.8 to 0.7, 0.6 and below 0.5 when $|G|$ increases from 2 to 5. Though *dissimulation* fails *d4* when $|G| \leq 3$, this strategy's advantage lies in its ability in solving multiple goals situation.

Combination is strictly better than *d4* in all cases. As a balanced strategy between *simulation* and *dissimulation*, two concepts of deception are both embedded in one strategy, and thus the merits of the two enable the *combination* strategy to outperform *d4* in different situations. For example, it has a similar performance to *simulation* when $|G| \leq 3$ and *dissimulation* when $|G| > 3$. With other concepts of deception suited for different problem domains or tasks coming in, *combination* or a similar balanced strategy using different weight settings, along with our magnitude-based deception maximization model could be of great help in planning suitable paths.

The experiments prove the effectiveness of our method in making full use of environment information, compared to the work in [5] that narrowly focuses on finding the tipping point (LDP) where the probability of the real goal becomes equal to that of some other goal. Overall, the results shown in Figure 6 prove that our single real goal magnitude-based deceptive path-planning has a decent, better, and more flexible performance.



(**a**) $|G| = 2$ (**b**) $|G| = 3$

(**c**) $|G| = 4$ (**d**) $|G| = 5$

Figure 6. The *F-measure* of the goal recognizer with deceptive paths generated using different strategies, at each interval t under different goal settings ($N_{stage} = 5$).

6. Conclusions

In this paper, we formalise deception as it applies to path-planning and present strategies for the generation of deceptive paths. The paper propose a single, real goal, magnitude-based deception maximization model which establishes a computable foundation for any further imposition of different deception strategies, and broadens the model's applicability in many scenarios. Different from the previous method [5] where deception could only be exploited from two goals (the real goal and a specific bogus goal), our method fully takes advantage of all possible goals during the generation of deceptive path. Also, the moving resource can be flexibly defined to serve the deceiver's trade-off between deceptivity and resource. As for the deceptive strategy, taking advantage of the model's flexibility, we propose a weighted combination strategy based on the original simulation and dissimulation. In general, our method shows much more flexibility and generality compared to the existing work from many aspects.

Though this work is quite suggestive, there are still many open problems that could be researched in the future. Firstly, the paper assumes that the defender is naive and rational. It is naive because the defender does not expect to be deceived, and thus deploys no counter plans or tactics. Its rationality is inherited from the goal recognition paradigm. If it is not, on one hand it may use machine learning approaches, e.g., the work in [70], to adapt its goal recognition model, or it would take more active counter behaviors; e.g., in the public security domain restricting the evader's actions at certain locations by using patrolling vehicles [34]. Further, it could be formulated in terms of a zero-sum game in which both players are trying to act optimally [83]. At this time, it would be more complicated to model these interactions between the two.

Secondly, the current model assumes that the environment and deceiver's actions are fully observable to the goal recognizer or the observer. However, for partially observable situations where its actions could be masked and behavioral patterns be covered, we would expect many more tactics could be available for the evader. Also as current formulation only considers one real goal setting, this work could be further extended to multiple real goal settings and thus transforming the problem into non-linear path-planning, which would be a much more valuable scenario where the evader have more than one targets.

Lastly, we believe that the concepts of deception, along with its models, could also be applied in other domains, such as classical planning using a plan library, etc. Similarly, a probabilistic plan recognition algorithm could then be used to evaluate the magnitude of deception, and a newly proposed optimization model with resource constraint to maximize the plan deceptivity.

Author Contributions: Conceptualization, K.X.; methodology, Y.Z.; formal analysis, L.Q.; supervision, Q.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially sponsored by the National Natural Science Foundation of China under Grants No.61473300, and Natural Science Foundation of Hunan Province under Grants No. 2017JJ3371.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
UAV	Unmanned Aerial Vehicle
LDP	Last Deceptive Point
AHMM	Abstract Hidden Markov Models
GRD	Goal Recognition Design
wcd	worst-case distinctiveness
RN	Road Network
SGPP	Single Goal Path-Planning
PGR	Probabilistic Goal Recognition
SRGDPP	Single Real Goal Deceptive Path-Planning
SRGMDPP	Single Real Goal Magnitude-based Deceptive Path-Planning
SRGMDM	Single Real Goal Magnitude-based Deception Maximization

References

1. Geib, C.W.; Goldman, R.P. Plan recognition in intrusion detection systems. In Proceedings of the DARPA Information Survivability Conference and Exposition II. DISCEX'01, Anaheim, CA, USA, 12–14 June 2001; Volume 1, pp. 46–55.
2. Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawa, E. Robocup: The robot world cup initiative. In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA, USA, 2–5 February 1997; pp. 340–347.
3. Root, P.; De Mot, J.; Feron, E. Randomized path planning with deceptive strategies. In Proceedings of the 2005, American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 1551–1556.
4. Keren, S.; Gal, A.; Karpas, E. Privacy Preserving Plans in Partially Observable Environments. In Proceedings of the IJCAI, New York, NY, USA, 9–15 July 2016; pp. 3170–3176.
5. Masters, P.; Sardina, S. Deceptive Path-Planning. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, Melbourne, Australia, 7–11 July 2017; pp. 4368–4375.
6. Masters, P. Goal Recognition and Deception in Path-Planning. Ph.D. Thesis, RMIT University, Melbourne, Australia, 2019.
7. Ramırez, M.; Geffner, H. Probabilistic plan recognition using off-the-shelf classical planners. In Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010), Atlanta, GA, USA, 11–15 July 2010; pp. 1121–1126.
8. Masters, P.; Sardina, S. Cost-based goal recognition for path-planning. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Sao Paulo, Brazil, 8–12 May, 2017; pp. 750–758.
9. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
10. Korf, R.E. Real-time heuristic search. *Artif. Intell.* **1990**, *42*, 189–211. [CrossRef]

11. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
12. Bui, H.H. A general model for online probabilistic plan recognition. *IJCAI* **2003**, *3*, 1309–1315.
13. Geib, C.W.; Goldman, R.P. Partial Observability and Probabilistic Plan/Goal Recognition. 2005. Available online: http://rpgoldman.real-time.com/papers/moo2005.pdf (accessed on 2 December 2019).
14. Sukthankar, G.; Geib, C.; Bui, H.H.; Pynadath, D.; Goldman, R.P. *Plan, Activity, and Intent Recognition: Theory and Practice*; Newnes: London, UK, 2014.
15. Whaley, B. Toward a general theory of deception. *J. Strateg. Stud.* **1982**, *5*, 178–192. [CrossRef]
16. Turing, A.M. Computing machinery and intelligence. In *Parsing the Turing Test*; Springer: New York, NY, USA, 2009; pp. 23–65.
17. Hespanha, J.P.; Ateskan, Y.S.; Kizilocak, H. Deception in Non-Cooperative Games with Partial Information. 2000. Available online: https://www.ece.ucsb.edu/hespanha/published/deception.pdf (accessed on 2 December 2019).
18. Hespanha, J.P.; Kott, A.; McEneaney, W. Application and value of deception. *Adv. Reason. Comput. Approaches Read. Opponent Mind* **2006**, 145–165.
19. Ettinger, D.; Jehiel, P. Towards a Theory of Deception. 2009. Available online: https://ideas.repec.org/p/cla/levrem/122247000000000775.html (accessed on 2 December 2019).
20. Arkin, R.C.; Ulam, P.; Wagner, A.R. Moral decision making in autonomous systems: Enforcement, moral emotions, dignity, trust, and deception. *Proc. IEEE* **2012**, *100*, 571–589. [CrossRef]
21. Alloway, T.P.; McCallum, F.; Alloway, R.G.; Hoicka, E. Liar, liar, working memory on fire: Investigating the role of working memory in childhood verbal deception. *J. Exp. Child Psychol.* **2015**, *137*, 30–38. [CrossRef]
22. Dias, J.; Aylett, R.; Paiva, A.; Reis, H. The Great Deceivers: Virtual Agents and Believable Lies. 2013. Available online: https://pdfs.semanticscholar.org/ced9/9b29b53008a285296a10e7aeb6f88c79639e.pdf (accessed on 2 December 2019).
23. Greenberg, I. The effect of deception on optimal decisions. *Op. Res. Lett.* **1982**, *1*, 144–147. [CrossRef]
24. Matsubara, S.; Yokoo, M. Negotiations with inaccurate payoff values. In Proceedings of the International Conference on Multi Agent Systems (Cat. number 98EX160), Paris, France, 3–7 July 1998; pp. 449–450.
25. Hausch, D.B. Multi-object auctions: Sequential vs. simultaneous sales. *Manag. Sci.* **1986**, *32*, 1599–1610. [CrossRef]
26. Yavin, Y. Pursuit-evasion differential games with deception or interrupted observation. *Comput. Math. Appl.* **1987**, *13*, 191–203. [CrossRef]
27. Hespanha, J.P.; Prandini, M.; Sastry, S. Probabilistic pursuit-evasion games: A one-step nash approach. In Proceedings of the 39th IEEE Conference on Decision and Control (Cat. number 00CH37187), Sydney, Australia, 12–15 December 2000; Volume 3, pp. 2272–2277.
28. Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; Meyer, G. Protect: A Deployed Game Theoretic System to Protect the Ports of the United States. 2012. Available online: https://www.ntu.edu.sg/home/boan/papers/AAMAS2012-protect.pdf (accessed on 2 December 2019).
29. Billings, D.; Papp, D.; Schaeffer, J.; Szafron, D. Poker as a Testbed for AI Research. In *Conference of the Canadian Society for Computational Studies of Intelligence*; Springer: New York, NY, USA, 1998; pp. 228–238.
30. Bell, J.B. Toward a theory of deception. *Int. J. Intell. Count.* **2003**, *16*, 244–279. [CrossRef]
31. Kott, A.; McEneaney, W.M. *AdversariaL Reasoning: Computational Approaches to Reading The Opponent'S Mind*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2006.
32. Jian, J.Y.; Matsuka, T.; Nickerson, J.V. Recognizing Deception in Trajectories. 2006. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.489.165rep=rep1type=pdf (accessed on 2 December 2019).
33. Shim, J.; Arkin, R.C. Biologically-inspired deceptive behavior for a robot. In *International Conference on Simulation of Adaptive Behavior*; Springer: New York, NY, USA, 2012; pp. 401–411.
34. Keren, S.; Gal, A.; Karpas, E. Goal Recognition Design. In Proceedings of the ICAPS, Portsmouth, NH, USA, 21–26 June 2014.
35. Keren, S.; Gal, A.; Karpas, E. *Goal Recognition Design for Non-Optimal Agents*; AAAI: Menlo Par, CA, USA, 2015; pp. 3298–3304.

36. Keren, S.; Gal, A.; Karpas, E. *Goal Recognition Design with Non-Observable Actions*; AAAI: Menlo Par, CA, USA, 2016; pp. 3152–3158.
37. Wayllace, C.; Hou, P.; Yeoh, W.; Son, T.C. Goal Recognition Design With Stochastic Agent Action Outcomes. In Proceedings of the IJCAI, New York, NY, USA, 9–15 July 2016.
38. Mirsky, R.; Gal, Y.K.; Stern, R.; Kalech, M. Sequential plan recognition. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Singapore, 9–13 May, 2016; pp. 1347–1348.
39. Almeshekah, M.H.; Spafford, E.H. Planning and integrating deception into computer security defenses. In Proceedings of the 2014 New Security Paradigms Workshop, Victoria, BC, Canada, 15–18 September 2014; pp. 127–138.
40. Lisỳ, V.; Píbil, R.; Stiborek, J.; Bošanskỳ, B.; Pěchouček, M. Game-theoretic approach to adversarial plan recognition. In Proceedings of the 20th European Conference on Artificial Intelligence, Montpellier, France, 27–31 August 2012; pp. 546–551.
41. Rowe, N.C. A model of deception during cyber-attacks on information systems. In Proceedings of the IEEE First Symposium onMulti-Agent Security and Survivability, Drexel, PA, USA, 31 August 2004; pp. 21–30.
42. Brafman, R.I. A Privacy Preserving Algorithm for Multi-Agent Planning and Search. In Proceedings of the IJCAI, Buenos Aires, Argentina, 25–31 July 2015; pp. 1530–1536.
43. Kulkarni, A.; Klenk, M.; Rane, S.; Soroush, H. Resource Bounded Secure Goal Obfuscation. In Proceedings of the AAAI Fall Symposium on Integrating Planning, Diagnosis and Causal Reasoning, New Orleans, LA, USA, 2–7 February 2018.
44. Kulkarni, A.; Srivastava, S.; Kambhampati, S. A unified framework for planning in adversarial and cooperative environments. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, FA, USA, 2–7 February 2018.
45. Kautz, H.A.; Allen, J.F. *Generalized Plan Recognition*; AAAI: Menlo Par, CA, USA, 1986; Volume 86, p. 5.
46. Pynadath, D.V.; Wellman, M.P. Generalized queries on probabilistic context-free grammars. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 65–77. [CrossRef]
47. Pynadath, D.V. *Probabilistic Grammars for Plan Recognition*; University of Michigan: Ann Arbor, MI, USA, 1999.
48. Pynadath, D.V.; Wellman, M.P. Probabilistic State-Dependent Grammars for Plan Recognition. 2000. Available online: https://arxiv.org/ftp/arxiv/papers/1301/1301.3888.pdf (accessed on 2 December 2019).
49. Geib, C.W.; Goldman, R.P. A probabilistic plan recognition algorithm based on plan tree grammars. *Artif. Intell.* **2009**, *173*, 1101–1132. [CrossRef]
50. Wellman, M.P.; Breese, J.S.; Goldman, R.P. From knowledge bases to decision models. *Knowl. Eng. Rev.* **1992**, *7*, 35–53. [CrossRef]
51. Charniak, E.; Goldman, R.P. A Bayesian model of plan recognition. *Artif. Intell.* **1993**, *64*, 53–79. [CrossRef]
52. Bui, H.H.; Venkatesh, S.; West, G. Policy recognition in the abstract hidden markov model. *J. Artif. Intell. Res.* **2002**, *17*, 451–499. [CrossRef]
53. Liao, L.; Patterson, D.J.; Fox, D.; Kautz, H. Learning and inferring transportation routines. *Artif. Intell.* **2007**, *171*, 311–331. [CrossRef]
54. Xu, K.; Xiao, K.; Yin, Q.; Zha, Y.; Zhu, C. Bridging the Gap between Observation and Decision Making: Goal Recognition and Flexible Resource Allocation in Dynamic Network Interdiction. In Proceedings of the IJCAI, Melbourne, Australia, 7–11 July 2017; Volume 4477.
55. Baker, C.L.; Saxe, R.; Tenenbaum, J.B. Action understanding as inverse planning. *Cognition* **2009**, *113*, 329–349. [CrossRef] [PubMed]
56. Ramırez, M.; Geffner, H. Plan recognition as planning. In Proceedings of the 21st international joint conference on Artifical intelligence, Pasadena, CA, USA, 11–17 July 2009; pp. 1778–1783.
57. Ramırez, M.; Geffner, H. Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 2009–2014.

58. Sohrabi, S.; Riabov, A.V.; Udrea, O. Plan Recognition as Planning Revisited. In Proceedings of the IJCAI, New York, NY, USA, 9–15 July 2016; pp. 3258–3264.
59. Albrecht, D.W.; Zukerman, I.; Nicholson, A.E. Bayesian models for keyhole plan recognition in an adventure game. *User Model. User-Adapt. Interact.* **1998**, *8*, 5–47. [CrossRef]
60. Goldman, R.P.; Geib, C.W.; Miller, C.A. A New Model of Plan Recognition. 1999. Available online: https://arxiv.org/ftp/arxiv/papers/1301/1301.6700.pdf (accessed on 2 December 2019).
61. Doucet, A.; De Freitas, N.; Murphy, K.; Russell, S. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, Stanford, CA, USA, 30 June–3 July 2000; pp. 176–183.
62. Saria, S.; Mahadevan, S. Probabilistic Plan Recognition in Multiagent Systems. 2004. Available online: https://people.cs.umass.edu/mahadeva/papers/ICAPS04-035.pdf (accessed on 2 December 2019).
63. Blaylock, N.; Allen, J. Fast Hierarchical Goal Schema Recognition. 2006. Available online: http://www.eecs.ucf.edu/gitars/cap6938/blaylockaaai06.pdf (accessed on 2 December 2019).
64. Singla, P.; Mooney, R.J. *Abductive Markov Logic for Plan Recognition*; AAAI: Menlo Park, CA, USA, 2011; pp. 1069–1075.
65. Yin, Q.; Yue, S.; Zha, Y.; Jiao, P. A semi-Markov decision model for recognizing the destination of a maneuvering agent in real time strategy games. *Math. Problems Eng.* **2016**, *2016*. [CrossRef]
66. Yue, S.; Yordanova, K.; Krüger, F.; Kirste, T.; Zha, Y. A Decentralized Partially Observable Decision Model for Recognizing the Multiagent Goal in Simulation Systems. *Discret. Dyn. Nat. Soc.* **2016**, *2016*. [CrossRef]
67. Min, W.; Ha, E.; Rowe, J.P.; Mott, B.W.; Lester, J.C. Deep Learning-Based Goal Recognition in Open-Ended Digital Games. *AIIDE* **2014**, *14*, 3–7.
68. Bisson, F.; Larochelle, H.; Kabanza, F. Using a Recursive Neural Network to Learn an Agent's Decision Model for Plan Recognition. Available online: http://www.dmi.usherb.ca/larocheh/publications/ijcai15.pdf (accessed on 2 December 2019).
69. Tastan, B.; Chang, Y.; Sukthankar, G. Learning to intercept opponents in first person shooter games. In Proceedings of the 2012 IEEE Conference on Computational Intelligence and Games (CIG), Granada, Spain, 11–14 September 2012; pp. 100–107.
70. Zeng, Y.; Xu, K.; Yin, Q.; Qin, L.; Zha, Y.; Yeoh, W. Inverse Reinforcement Learning Based Human Behavior Modeling for Goal Recognition in Dynamic Local Network Interdiction. In Proceedings of the AAAI Workshops on Plan, Activity and Intent Recognition, New Orleans, LA, USA, 2–7 February 2018.
71. Agotnes, T. Domain Independent Goal Recognition. Stairs 2010: Proceedings of the Fifth Starting AI Researchers Symposium. 2010. Available online: http://users.cecs.anu.edu.au/ssanner/ICAPS2010DC/Abstracts/pattison.pdf (accessed on 2 December 2019).
72. Pattison, D.; Long, D. Accurately Determining Intermediate and Terminal Plan States Using Bayesian Goal Recognition. In Proceedings of the ICAPS, Freiburg, Germany, 11–16 June 2011.
73. Yolanda, E.; R-Moreno, M.D.; Smith, D.E. A Fast Goal Recognition Technique Based on Interaction Estimates. 2015. Available online: https://www.ijcai.org/Proceedings/15/Papers/113.pdf (accessed on 2 December 2019).
74. Pereira, R.F.; Oren, N.; Meneguzzi, F. Landmark-based heuristics for goal recognition. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017.
75. Cohen, P.R.; Perrault, C.R.; Allen, J.F. Beyond question answering. *Strateg. Nat. Lang. Process.* **1981**, *245274*.
76. Jensen, R.M.; Veloso, M.M.; Bowling, M.H. OBDD-Based Optimistic and Strong Cyclic Adversarial Planning. 2014. Available online: https://pdfs.semanticscholar.org/59f8/fd309d95c6d843b5f7665bbf9337f568c959.pdf (accessed on 2 December 2019).
77. Avrahami-Zilberbrand, D.; Kaminka, G.A. Keyhole adversarial plan recognition for recognition of suspicious and anomalous behavior. *Plan Activ. Int. Recognit.* **2014**, 87–121.
78. Avrahami-Zilberbrand, D.; Kaminka, G.A. *Incorporating Observer Biases in Keyhole Plan Recognition (Efficiently!)*; AAAI: Menlo Park, CA, USA, 2007; Volume 7, pp. 944–949.
79. Braynov, S. Adversarial Planning and Plan Recognition: Two Sides of the Same Coin. 2006. Available online: https://csc.uis.edu/faculty/sbray2/papers/SKM2006.pdf (accessed on 2 December 2019).
80. Le Guillarme, N.; Mouaddib, A.I.; Gatepaille, S.; Bellenger, A. Adversarial Intention Recognition as Inverse Game-Theoretic Planning for Threat Assessment. In Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), San Jose, CA, USA, 6–8 November 2016; pp. 698–705.

81. Lofberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In Proceedings of the CACSD Conference, New Orleans, LA, USA, 2–4 September 2005; pp. 284–289.
82. Sturtevant, N.R. Benchmarks for grid-based pathfinding. *IEEE Trans. Comput. Intell. AI Games* **2012**, *4*, 144–148. [CrossRef]
83. Xu, K.; Yin, Q. Goal Identification Control Using an Information Entropy-Based Goal Uncertainty Metric. *Entropy* **2019**, *21*, 299. [CrossRef]