Deceptive Path-Planning

Peta Masters and **Sebastian Sardina**
RMIT University, Melbourne, Australia
{peta.masters, sebastian.sardina}@rmit.edu.au

Abstract

Deceptive path-planning involves finding a path such that the probability of an observer identifying its final destination—before it has been reached—is minimised. This paper formalises deception as it applies to path-planning and introduces the notion of a last deceptive point (LDP) which, when measured in terms of *path completion*, can be used to rank paths by their potential to deceive. Building on recent developments in probabilistic goal-recognition, we propose a formula to calculate an optimal LDP and present strategies for the generation of deceptive paths by both simulation ('showing the false') and dissimulation ('hiding the real').

1 Introduction

In this paper, we study deceptive path-planning (DPP): the problem of finding a path through a map (or model) of a domain such that an observer, watching an agent make her way along the path, will be unable to determine—until the last possible moment—where the agent is going.

This work lies at the intersection of two well-established and much-studied disciplines within Computer Science: *path-planning*, which is the problem of finding a path through a graph or grid from a given start location to a given goal [Hart *et al.*, 1968; Korf, 1990; LaValle, 2006]; and *goal recognition*, which tries to determine an agent's purpose by observing her actions [Kautz and Allen, 1986; Charniak and Goldman, 1991; Ramirez and Geffner, 2010].

The deception problem is significant. It is a topic with a long history in Computer Science, particularly within the realms of Artificial Intelligence [Turing, 1950] and game theory [Hespanha *et al.*, 2000], and one of increasing relevance in social robotics [Arkin *et al.*, 2012], where interactions between autonomous situated agents and (sometimes vulnerable) humans may usefully, it is argued, include deceptive practices [Sharkey and Sharkey, 2012; Shim and Arkin, 2013]. Deception is a key indicator for intelligence [Alloway *et al.*, 2015]; furthermore, agents and game characters who lie and cheat are more believable, more interesting and more fun to play against than those that play fair [Dias *et al.*, 2013].

When *advocating* deception, the literature sometimes prefers to characterise it as 'privacy protection' [Keren *et al.*, 2016]. Consider a convoy escorting a VIP to one of five possible destinations. An observer plans to deploy an assassin once the VIP's destination is known. What (deceptive) path will (protect her privacy so as to) minimise the likelihood of the observer correctly identifying the convoy's destination?

We base our definitions on a general theory of deception [Bell, 2003; Whaley, 1982; Bowyer, 1982], which describes the concept in terms of *simulation* (showing the false) and *dissimulation* (hiding the true). Technically, though, our work builds on—and inverts—recent model-based approaches to goal recognition [Ramirez and Geffner, 2010]. In particular, we depend on the ability to determine *the probability of each goal at any given point along a path.* Informally, the solution to a probabilistic goal recognition problem is a probability distribution that ranks goals by their likelihood. Our thesis is based on the simple proposition that goals ranked for likelihood can similarly be ranked for *un*likelihood. Once we know how unlikely an observer believes the *real* goal to be, we have a measure by which to assess how successfully she has been deceived.

To quantify path deceptivity, we measure its *magnitude* (at each step), *density* (number of steps) and *extent* (distance travelled). We define all three concepts but, in this paper, focus particularly on extent. We introduce the notion of a 'last deceptive point' (LDP) and present a novel way of measuring its location using 'path-completion'. This method (as distinct from actual path length or counting steps [Keren *et al.*, 2015]) enables us to identify a position in a suboptimal path *without having to place budgetary limits on its length or cost.*

To compute deceptive paths, we exploit the recent finding that, given an agent's starting point and current location, goal recognition for path-planning can be achieved without reference to any other observation [Masters and Sardina, 2017]. That is, a probability distribution can be precalculated at any node and remains constant irrespective of the path taken to reach it. We present a formula to calculate the radius within which the probability of a goal dominates and show that, when it corresponds to the LDP of a path, the deceptivity of that path, in terms of extent, is optimal within the domain.

In the rest of this paper, we set out the background, then formally define deception in terms of path-planning. We show how to maximise the LDP and present strategies for computing deceptive paths. Finally, we provide an empirical evaluation, review related work and present our conclusion.

2 Background

Path-planning is the problem of finding a path from a given initial location to a given final destination in a given map (or model) of a domain. It is a problem with multiple applications, from robotics [Siegwart *et al.*, 2011] and video games [Millington and Funge, 2009] to road network navigation [Bast *et al.*, 2015].

Formally, a ***path-planning domain*** is a triple $D = \langle N, E, c\rangle$, where:

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

A ***path*** π through a path-planning domain is a sequence of nodes $\pi = n_0, n_1, .., n_k$ such that $(n_i, n_{i+1}) \in E$ for each $i \in \{0, 1, .., k-1\}$. We use π^i to denote the i-th node in π, and $|\pi|$ to denote the length of π, being the total number of edges in π, that is, $\pi^{|\pi|} = n_k$. The ***cost*** of π is the cost of traversing all edges in π, that is, $\mathit{cost}(\pi) = \sum_{i=0}^{k-1} c(\pi^i, \pi^{i+1})$. The ***set of all paths*** in the domain is denoted by Π, and the set of all paths π starting at $\pi^0 = n_1$ and ending at $\pi^{|\pi|} = n_2$ is denoted by $\Pi(n_1, n_2)$.

A ***path-planning problem*** is a tuple $\langle D, s, g\rangle$, where:

- $D = \langle N, E, c\rangle$ is a path-planning domain;
- $s \in N$ is the start location; and
- $g \in N$ is the goal/destination location.

The solution to a path-planning problem or ***solution path*** is a path π in the corresponding domain D such that $\pi^0 = s$ and $\pi^{|\pi|} = g$; the set of all of them being $\Pi(s, g)$. An ***optimal path*** is a solution path with the lowest cost among all solution paths. The ***optimal cost*** between two nodes is the cost of an optimal path between them and the optimal cost from n_i to n_j is denoted by $\mathit{optc}(n_i, n_j)$. To find an optimal path, typical AI approaches use advanced variations of A^* [Hart *et al.*, 1968], a well-known best-first search algorithm.

In this paper, domain costs are synonymous with distance: when we speak of 'radius', it is properly a 'cost radius'.

2.1 Goal Recognition in Path-Planning

The act of deception implies the existence of an observer to be deceived; deception *by path-planning* implies an observer who is *trying to work out where the path is going*, such as the assassin's controller in our motivating example. This is the domain of goal or plan recognition:[1] the problem of identifying an agent's goal or intent by observing her actions.

A ***probabilistic goal recognition problem*** (in a path-planning domain) is a tuple $\langle D, G, s, O, \mathit{Prob}\rangle$, where:

- $D = \langle N, E, c\rangle$ is a path-planning domain;
- $G \subseteq N$ is the set of possible goal locations;
- $s \in N$ is the start location;
- $O = o_1 \cdots o_k \in N^*$, where $k > 0$, is a sequence of observations (which may or may not be adjacent); and
- *Prob* represents the prior probabilities of the goals (though we assume that priors for all goals are equal).

A solution is a (conditional) probability distribution across G: $Pr(g|O)$ denotes the probability of location $g \in G$ being the destination, given the observations O. We use $g_r \in G$ to denote the ***real goal*** (i.e., the goal which the agent is targeting and at which the path will eventually terminate). The quality of a solution, then, is measured by its success in determining that $Pr(g_r|O) \geq Pr(g|O)$ for all $g \in G \setminus \{g_r\}$.

Traditionally, goal recognition has relied on a library of pre-existing plans against which to match observations [Kautz and Allen, 1986; Charniak and Goldman, 1991], the idea being that, having identified the plan, you have implicitly identified the goal [Demolombe and Hamon, 2002].

A recently introduced alternative technique, based on the assumption that a rational agent is taking the optimal (or 'least suboptimal') path to goal, dispenses with the plan library and instead associates the probability of a plan with its cost [Baker *et al.*, 2009; Ramirez and Geffner, 2009]. We focus on the probabilistic approach taken in [Ramirez and Geffner, 2010] whereby an off-the-shelf planning system is used to determine (for each goal) the ***cost difference*** between the cheapest plan that incorporates the observations and the cheapest plan that (at least partially) avoids them.

$$\mathit{costdif}_{RG}(s, g, O) = \mathit{optc}(s, O, g) - \mathit{optc}^{\neg}(s, O, g)$$

By comparing cost differences across goals, a probability distribution is generated that conforms to the intuition: that is, the lower the cost difference, the higher the probability.

In grounding the Ramirez and Geffner formula to path-planning, Masters and Sardina [2017] achieve an almost identical probability distribution[2] without reference to any but the final observation in the sequence:

$$\mathit{costdif}(s, g, n) = \mathit{optc}(n, g) - \mathit{optc}(s, g), \qquad \text{(MS1)}$$

where s is the start, g is a goal and $n = O^{|O|}$ (that is, n is the most recently observed/current location of the agent whose destination we wish to determine). The posterior probability distribution $P(G|N)$ retains the essential property that the lower the cost difference, the higher the probability:

Lemma 1 *$P(g|n) > P(g'|n)$ if and only if $\mathit{costdif}(s, g, n) < \mathit{costdif}(s, g', n)$.*

The impact of Equation (MS1) is that the probability distribution at any given location remains constant, regardless of the path that led to it (or how often it is revisited) and can be precalculated even before an agent enters the domain. This allows for precalculation of a sort of 'heat-map' showing the probability of each goal at any/every location. Furthermore, by means of the heat-map, it is possible to plot a perimeter around each goal within which its probability exceeds that of any other goal in the domain.

In Section 4, we will see that the deceptivity of a path is constrained by this perimeter; and in Section 4.2, we present a simple formula to calculate its closest distance from goal.

[1] Goal recognition is a subproblem of plan recognition. The literature uses both terms and we use them interchangeably in this text.

[2] The ordering of goals by probability differs *only* if all optimal paths must incorporate the observations, which, as noted by Ramirez and Geffner, is arguably a non-realistic case.

3 Deceptive Path-Planning

In this section, we take definitions of deception and related concepts from the fields of social science and diplomatic strategy and formalise them in the context of path-planning.

Whereas path-planning finds a path (typically the least expensive path) to a goal and goal recognition attempts to identify which, in a set of possible goals, an observed path is targeting, *deceptive path-planning* finds a path to goal that minimises the likelihood of an observer identifying which, in a set of goals—such as the multiple possible destinations of our VIP's convoy—is being targeted.

Definition 1 *A* ***deceptive path-planning*** *(*DPP*) problem is a tuple* $\langle\langle D, s, g_r\rangle, G, P\rangle$*, where:*

- $\langle D, s, g_r\rangle$ *is a path-planning problem, with* $D = \langle N, E, c\rangle$ *being its path-planning domain,* $s \in N$ *the start location and* $g_r \in G$ *the real goal;*
- $G \subseteq N$ *is the set of possible goal locations; and*
- $P(G|O \cdot n)$ *denotes the posterior probability of a goal given a sequence of observations (or last node in that sequence). That is,* P *stands for the model of the observer.*

The solution to a DPP problem is a solution to its path-planning problem that is 'deceptive'. The quality of the solution depends on the magnitude, density, and extent of the deception, as we now discuss.

Deception is the "distortion of perceived reality" [Bowyer, 1982, p.47] and may be achieved in one of two ways: by *simulation* ('showing the false') or *dissimulation* ('hiding the real') [Bell, 2003; Whaley, 1982]. In path-planning, the only reality is movement towards a goal. In this context, 'hiding the real' and 'showing the false' equate to obscuring the path-planner's true destination and/or creating the impression that she is going somewhere that she is not.

3.1 A Deceptive Step

We first examine deception as it applies to an individual node or step along the path. It is at this level that we assess *deceptive magnitude*, captured here in two distinct notions of simulation and dissimulation.

Definition 2 *A* ***truthful*** *step is a node at which the probability of the real goal* g_r *is greater than the probability of any other possible goal, that is,* $P(g_r|O \cdot n) > P(g|O \cdot n)$*, for all* $g \in G \setminus \{g_r\}$*. Otherwise, the step is* ***deceptive****.*

Definition 3 ***Simulation*** *(showing the false) occurs when the probability of a bogus goal is strictly greater than the probability of the real goal* g_r*, that is, there exists* $g \in G \setminus \{g_r\}$ *such that* $P(g_r|O \cdot n) < P(g|O \cdot n)$*.*

We quantify simulation by measuring the amount by which a false goal dominates the real goal. The greater the dominance, the greater the deception.

$$\textit{simulation}(O{\cdot}n) = \max_{g_i \in G\setminus\{g_r\}} P(g_i|O{\cdot}n) - P(g_r|O{\cdot}n). \quad (1)$$

If we simulate successfully, our hypothetical assassin is deployed to the wrong location; the VIP survives.

Definition 4 ***Dissimulation*** *(hiding the real) occurs when the probability of the real goal* g_r *is less than or equal to the probability of another goal, that is, there exists* $g \in G \setminus \{g_r\}$ *such that* $P(g_r|O \cdot n) \leq P(g|O \cdot n)$*.*

Following Bowyer [1982], our definition of deception *always* involves dissimulation and *may also* involve simulation. We quantify that aspect of deception exclusive to dissimulation (the degree of ambiguity) using Shannon's entropy:

$$\textit{dissimulation}(O \cdot n) = \sum_{g_i \in G} P(g_i|O \cdot n) \times log_2(P(g_i|O \cdot n)). \quad (2)$$

If we dissimulate successfully, the controller does not know where to send the assassin; though she may guess correctly.

3.2 A Deceptive Path

In life, a deception might not be uncovered for months or years after it occurs (if ever); but a deceptive path, with full observability, is always ultimately truthful because the final step always arrives—and is seen to terminate—at its goal.[3] Thus, in a multi-goal domain (assuming equal priors) every path is deceptive at its start and truthful at its goal. It follows that, in every such path, there is one truthful node prior to which all previous steps (if any) are deceptive; and one deceptive node beyond which all subsequent steps (if any) are truthful. We call these the *first truthful point* (FTP) and *last deceptive point* (LDP) respectively.

Definition 5 *Given a path* π*, its* ***first truthful point*** FTP$_\pi$ *is a node* π^i*, which is itself truthful whereas all (if any) previous nodes* π^j*, for all* $j \in \{0, \ldots, i-1\}$*, are deceptive.*

Definition 6 *Given a path* π*, its* ***last deceptive point*** LDP$_\pi$ *is a node* π^i*, which is itself deceptive whereas all (if any) subsequent nodes* π^j*, for all* $j \in \{i{+}1, \ldots, |\pi|\}$*, are truthful.*

Depending on the relative location of these two points, we can identify two extreme notions of a deceptive path.

Definition 7 *A* ***strongly deceptive*** *path* π *is continually deceptive to its* LDP*, that is, if* LDP$_\pi = \pi^i$*, then* FTP$_\pi = \pi^{i+1}$*.*

Definition 8 *A* ***weakly deceptive*** *path* π *includes truthful steps before its* LDP*, that is, if* LDP$_\pi = \pi^i$*, then* FTP$_\pi = \pi^j$*, for some* $j < i$*.*

To assess the relative strengths of two paths, we measure their *deceptive density*. Clearly, we want to minimise the opportunities for an observer to correctly identify the real goal; that means minimising the number of truthful steps. The fewer such steps a path π contains, the greater its deceptive density:

$$\textit{density}(\pi) = \frac{1}{|N_t|}, \quad (3)$$

where N_t is the set of all truthful steps in π.

In the next section, we will use the LDP to measure a path's deceptive extent. Before proceeding, we briefly mention the potentially confusing relationship between path optimality and deceptivity. Note that a path can be truthful without being optimal (it may favour the real goal more than any bogus goal but still be a suboptimal path) and deceptive without being suboptimal (it may be an optimal path to multiple goals).

[3] Implicitly, we assume an observation at the *final time-step+1*, at which the agent is seen to remain—i.e., terminate—at her goal.

4 Maximising the Last Deceptive Point

As discussed, we measure a path's deceptivity at three levels of granularity: its LDP tells us the *extent* of the deception; the number of truthful steps determines *density*; and the degree of simulation or dissimulation per step gives us its *magnitude* (though full discussion of the latter is reserved for future work). In this section, we demonstrate a novel method of assessing distance travelled along a suboptimal path; then, assuming an observer consistent with the Ramirez and Geffner model, we present a formula to calculate the distance from goal within which *every* path is truthful.

4.1 Path Completion

The LDP is significant because it represents the point in a path at which a rational observer ceases to be deceived; that is, the moment when, in the eyes of the observer, the probability of the real goal comes to outweigh the probability of any other possible goal and beyond which the probability of that real goal dominates continuously until the goal is reached.

Intuitively, we want to delay the LDP until *as late in the path as possible*; but although the LDP is easy to identify, expressing its location in terms that allow for comparison between paths of different lengths and shapes is problematic.

Example 1 *Consider two paths $\pi_a, \pi_b \in \Pi(s, g_r)$ of different lengths taking entirely different routes from s to g_r. Suppose that LDP_{π_a} occurs at π_a^{312} and LDP_{π_b} at π_b^{203}. LDP_{π_a} is clearly 'later' if we are counting steps; but does that make π_a the more deceptive path? Suppose that $|\pi_a| - 312 = 200$ and $|\pi_b - 203| = 10$, that is, there are many more truthful steps from LDP_{π_a} to g_r than from LDP_{π_b} to g_r. This means an agent on π_b will be much closer to the goal before the path stops being deceptive. So, which LDP is really 'later'?*

In the absence of a cost or time constraint, a deceptive path may loop or backtrack indefinitely, delaying the LDP without making any progress towards the goal or, as in Example 1, causing an agent to traverse *more* truthful steps, not less! We therefore propose to ignore the actual path length and instead focus on the position of a node in terms of how much 'true work' has been completed towards the end goal.

If a path can be said to have a purpose, then its purpose is its goal. To capture 'true work', we define the notion of ***path completion***, which uses optimal cost (from s to g) to calculate how close the meandering suboptimal path π has come to achieving its purpose/reaching its goal g when it reaches a node n. Formally:

$$pcomp(n, s, g) = optc(s, g) - optc(n, g). \quad (4)$$

Figure 1 depicts this equation graphically. This concept is similar to 'task completion' in a project plan. No matter what resources or how many wrong, unnecessary and/or costly subtasks have been attempted, task completion is ultimately based on how much is left still to do.

When we measure the position of the LDP in terms of path completion, we learn how far an agent has been able to travel deceptively and (therefore) how far remains to be travelled 'in plain sight'. It is a method that explicitly enables us to distinguish between 'deceiving longer' (e.g., a path that repeatedly circles a bogus goal) and 'deceiving further' (i.e., a path that gets close to the real goal before it stops being deceptive).

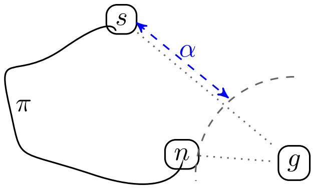

Figure 1: Path completion α based on optimal path (Equation 4). Straight lines represent optimal paths.

In the remainder of this paper, we make deceptive extent—and therefore path completion at the LDP—our primary focus in planning a deceptive path. So, we say that we have *maximised the last deceptive point* when $pcomp(\text{LDP}_\pi, s, g) \geq pcomp(\text{LDP}_{\pi_i}, s, g)$, for all $\pi_i \in \Pi(s, g)$.

4.2 The Radius of Maximum Probability

So far our framework has been agnostic with regard to the model of the observer (denoted by P in a deceptive path-planning task). From this point forward, however, we focus on a particular observer as represented by the goal recognition model in [Ramirez and Geffner, 2010].

Recall from Section 2.1 that, assuming an observer consistent with that model, the probability distribution across goals at each node remains constant, regardless of the path taken [Masters and Sardina, 2017]. Following this approach, all probabilities could be precalculated to create a 'heat-map' which would reveal the perimeter within which any given goal becomes 'most probable'. We identify the *minimum distance from such a perimeter to a potential goal g* as follows.

Definition 9 *The* ***radius of maximum probability*** *for a possible goal g, denoted RMP_g, is $x \in \mathbb{R}$ such that:*

1. *for all $n \in N$ such that $optc(n, g) < x$, it is the case that $P(g|n) > P(g'|n)$, for all $g' \in G \setminus \{g\}$; and*
2. *there exists a node $n' \in N$ such that $optc(n', g) = x$ and $P(g|n') \leq P(g'|n')$ for some $g' \in G \setminus \{g\}$.*

The RMP at the *real* goal, RMP_{g_r}, is of particular interest. It occurs at the minimum distance between a deceptive node and the real goal, and therefore represents a constraint, *imposed by the domain*, on the maximum value of any path's LDP.

Theorem 1 *Let π be a path such that $\pi \in \Pi(s, g_r)$. Then, $optc(\text{LDP}_\pi, g_r) \geq \text{RMP}_{g_r}$.*

Proof. By Definition 9, for all $n \in N$ such that $optc(n, g_r) < \text{RMP}_{g_r}$, n is truthful. But LDP_π is deceptive. Therefore $optc(\text{LDP}_\pi, g_r) \not< \text{RMP}_{g_r}$ and the proposition follows. □

That is, LDP_π *cannot* lie within the real goal's radius of maximum probability.

Corollary 1 *An LDP at RMP_{g_r} from g_r cannot be exceeded: if $\text{RMP}_{g_r} = optc(\text{LDP}_\pi, g_r)$, then $pcomp(\text{LDP}_\pi, s, g_r) \geq pcomp(\text{LDP}_{\pi_i}, s, g_r)$, for all $\pi_i \in \Pi(s, g_r)$.*

Thus, if we know the value of the RMP_{g_r}, we know the maximum LDP for the domain, namely:

$$\max_{\pi \in \Pi(s, g_r)} pcomp(\text{LDP}_\pi, s, g_r) = optc(s, g_r) - \text{RMP}_{g_r}. \quad (5)$$

A closed formula to calculate a lower bound for RMP_{g_r}

A naive way to compute RMP_{g_r} and identify a node at which the LDP can be maximised (i.e., a node at RMP_{g_r} from g_r) would be to perform a sort of breadth first search radiating out from the real goal, computing probabilities at every node until the first deceptive node has been reached. Taking the optimal cost from that node to goal gives us RMP_{g_r}. This is computationally onerous. Instead, we propose a simple formula that will enable us to calculate a theoretical radius within which we can guarantee that no deceptive node occurs; and we do this without having to calculate any probabilities whatsoever.

Intuitively, RMP_{g_r} signals a tipping point, where the probability of g_r becomes equal to the probability of some other goal $g' \in G \setminus \{g_r\}$, that is, a point where probabilities 'flip' from favouring one goal to favouring another. Since such probabilities are based on cost difference, a (hypothetical) node n at that tipping point will have the property, by Lemma 1, that $\textit{costdif}(s, g_r, n) = \textit{costdif}(s, g', n)$.

We want to find, in a convenient way, how far (in terms of cost) that hypothetical node n is from g_r and, if it exists, the node itself. Referring to Figure 2, we are constructing a formula for β. There, $a = \textit{optc}(s, g_r)$, $b = \textit{optc}(s, g')$, $c = \textit{optc}(g_r, g')$, $y = c - \beta$. The tipping point (node n) ought to be located on an optimal path from g_r to g', at a point where $\textit{optc}(n, g_r) = \beta$ and $\textit{optc}(n, g') = y$. Using Equation (MS1), we get:

$$\begin{aligned}
\textit{costdif}(s, g_r, n) &= \textit{costdif}(s, g', n) \\
\beta - a &= y - b \\
y &= \beta + b - a \\
c &= \beta + (\beta + b - a) \qquad \text{from } y = c - \beta \\
&= 2\beta + b - a \\
\beta &= \frac{c + a - b}{2}
\end{aligned}$$

This gives us β, the optimal cost from g_r to the point at which the hypothetical node n must occur. Since the domain may include many possible goals, we take the minimum (recall s is the starting point and G the set of possible goals):

$$\beta_{\min} = \min_{g_i \in G} \frac{\textit{optc}(g_r, g_i) + \textit{optc}(s, g_r) - \textit{optc}(s, g_i)}{2}. \quad (6)$$

So, $\beta_{\min}$ represents the optimal cost from g_r to the point at which $n_{\min}$ must occur (w.r.t. some possible goal $g_{\min}$). However, due to the discrete nature of our setting, the (precise) node $n_{\min}$, where the probabilities for g_r and some other goal $g_{\min}$ coincide, may not actually exist! If it does, then $\beta_{\min} = \textit{optc}(n_{\min}, g_r) = \text{RMP}_{g_r}$; otherwise $\beta_{\min}$ represents a (useful) lower bound for RMP_{g_r}.

The above equation is significant because it frees us from calculating probabilities at any particular node (or any node at all). We now use it to identify deceptive 'target nodes' at which an LDP can be maximised.

Definition 10 *A **target node** $t \in N$ is a deceptive node such that $\textit{optc}(t, g_r) \approx \beta_{\min}$.*[4]

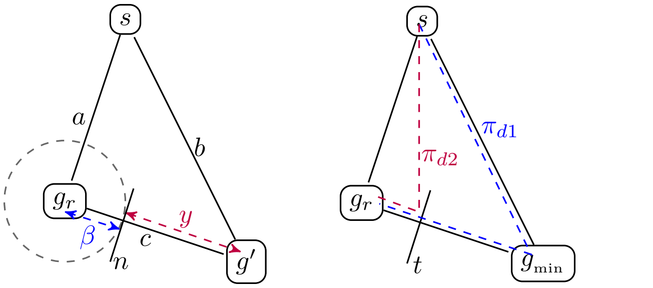

Figure 2: Labels a, b, and c stand for optimal costs between nodes.

Note that not all (and perhaps few) nodes at RMP_{g_r} cost distance from g_r are deceptive. Referring to Figure 2, however, we can readily identify one such node by retreating along (g_r, g') through the radius β from g_r (where a node *may* exist) to the first *actual* node on that edge.

In Section 5, we refer to this target node t and to $g_{\min}$ (the goal referenced by Equation 6), that is:

$$g_{\min} = \operatorname*{argmin}_{g_i \in G} \frac{\textit{optc}(g_r, g_i) + \textit{optc}(s, g_r) - \textit{optc}(s, g_i)}{2}. \quad (7)$$

5 Deceptive Path-Planning Strategies

Here we discuss approaches to the computation of paths whose deceptivity is maximised in terms of extent. We consider how the two fundamental deception strategies can help to maximise a path's deceptive density and minimise its cost.

Simulation. The simplest simulation (considered by Keren *et al.* [2015] w.r.t. "bounded deception") first takes an optimal path towards a bogus goal. Referring to Figure 2, this strategy generates $\pi_{d1} = s, .., g_{\min}, ..., g_r$. Computationally inexpensive, this achieves a *strongly* deceptive path (every step to the LDP is deceptive), and maximises deceptive density and extent (its LDP occurs at t). However, path cost is likely to be high and, although it initially deceives both human and automated observers, reaching but not stopping at $g_{\min}$ immediately signals to a human that $g_{\min}$ is *not* the real goal; though an automated observer utilising, for example, the [Ramirez and Geffner, 2010] model, is deceived all the way to t.

Dissimulation Dissimulation seeks an ambiguous path. The simplest such strategy takes an optimal path π_{d2} direct from s to t, then on to g_r. This generates the cheapest path that can pass through t. It *might* be deceptive to a human observer until later in the path than π_{d1} above. However, we would expect π_{d2} to be only *weakly* deceptive, that is, truthful steps are likely to occur before the LDP without additional checks and balances. We therefore propose two refinements:

1. A path π_{d3} can be assembled using a modified heuristic so that, while still targeting t, whenever there is a choice of routes, it favours the bogus goal, increasing its likelihood of remaining deceptive. Still using an off-the-shelf path-planner, the usual heuristic $h(n, t)$—which returns

[4] If there is no node at precisely $\beta_{\min}$, we take an approximation (always greater), and retreat to the closest *actual* node.

the estimated cost from node n to target h—is modified to also evaluate heuristics for g_r and $g_{\min}$:

$$\text{if } h(n, g_r) < h(n, g_{\min}) \text{ then } h(n,t) = \alpha h(n,t),$$

where constant $\alpha > 1$. Path π_{d3} is computationally more demanding than π_{d1} or π_{d2} (it evaluates more heuristics), but aims to approach π_{d1}'s deceptive density at something close to π_{d2}'s cost.

2. As an alternative—and perhaps definitive refinement with respect to cost—we can use a precalculated 'heat-map' of probabilities [Masters and Sardina, 2017] (or calculate them on-the-fly) to prune truthful nodes in the search. The resulting path π_{d4} is strongly deceptive with maximised LDP and maximum density at minimum cost.

   Contrary to the common idea of a deceptive path—that is, a rambling suboptimal path, full of expensive loops and unnecessary detours—this brute force strategy demonstrates that there is such a thing as a fully deceptive path that is also fully rational.

Differences between strategies are highlighted at Figure 3.

5.1 Experimental Evaluation

Though not proposed as optimised algorithms, we evaluated the relative efficiencies (time/cost) of the above strategies and the effectiveness (deceptivity) of paths they can produce. We generated a problem set based on game maps from the Moving-AI benchmarks [Sturtevant, 2012] to which we added three extra candidate goals at random locations. For each of 50 problems, we generated one optimal path using a standard implementation of A* and four deceptive paths (each using a different strategy). We timed path generation and recorded path costs. We truncated paths at the RMP (beyond which all paths would be truthful) and, using Ramirez and Geffner's method of goal recognition, calculated probabilities at intervals to confirm/assess deceptive density and extent.

Figure 4 captures our results. Comparison with A* shows a clear trade-off between cost and deceptivity. Strategies π_{d2} and π_{d3} returned comparatively cheap paths and were computationally efficient but performed erratically, each showing an increase in the number of paths deceptive immediately before the RMP. This implies they are only *weakly* deceptive (i.e., deceptive nodes may follow truthful ones). Simulation (π_{d1}) was strongly deceptive but the least efficient strategy in terms of cost. Dissimulation with pruning (π_{d4}) was fully deceptive at much lower cost. (In fact, as we know, paths generated using this strategy are optimal amongst deceptive paths.) Although generation of π_{d4} was slow, this is only because we calculated each node's deceptivity on the fly. If repeatedly considering deceptivity in a known domain, a probability heat-map could be precalculated (as previously discussed), enabling truthful nodes to be pruned in constant time, so closing the time difference between this and other strategies.

6 Related Work

Deceptive paths appear in the literature under various guises, though not previously (to our knowledge) as the solution to a path-planning problem.

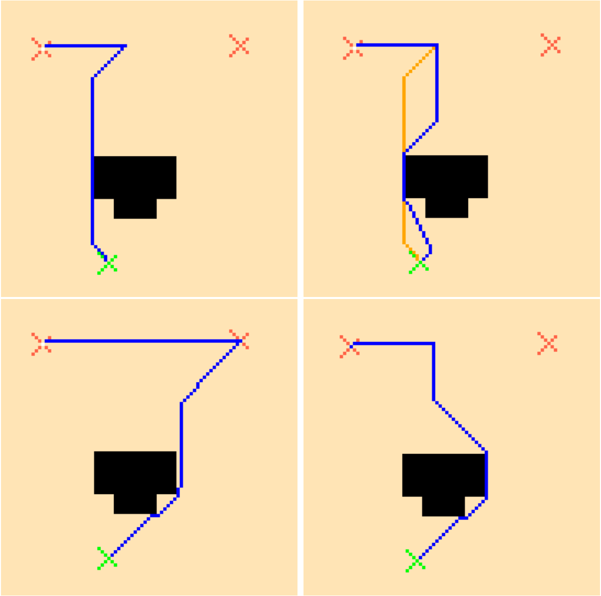

Figure 3: Deceptive path-planning strategies. Clockwise from bottom-left: paths π_{d1}, π_{d2}, π_{d3}, π_{d4}. Path π_{d3} is superimposed on π_{d2} to highlight the differences. Paths π_{d1} and π_{d4} both have maximum deceptive density but π_{d4} is optimal (amongst deceptive solutions that pass through t) with respect to cost.

Jian *et al.* [2006] conducted a pencil and paper study to find out if deceptivity could be *detected* from a path-plan. Subjects were asked to assume that they were under surveillance while drawing a path, from start to goal, without giving away their true destination. The study found 38 recognisable strategies, including "straight towards decoy", our baseline simulation strategy.

In his game theoretical account, Hespanha [2006] suggests that, when an observer does not know what to believe, she must make her decision as if she had made no observations at all. This is precisely the objective of a dissimulation strategy and one successfully exploited in an experiment by Root *et al.* [2005] in which drones conduct reconnaissance while under surveillance. In a domain modelled as a graph, the system selects a ground path, then constructs a set of flight plans that involve overflying not only that path but *every* edge capable of supporting military traffic. As predicted, execution of the paths renders observation meaningless: the defender must select from multiple routes, all with the same probability.

Simulation arises in a path-planning-related experiment carried out by roboticists Shim and Arkin [2012], inspired by the food-hoarding behaviour of squirrels. Computerised 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*) [Keren *et al.*, 2014], which involves modifying a domain's

	Path cost	Gen. time	10%	25%	50%	75%	90%	99%
π_{A^*}	215.9	0.208	78	68	40	32	22	12
π_{d1}	375.2	1.378	**100**	**100**	**100**	**100**	**100**	**100**
π_{d2}	245.2	1.997	92	88	76	72	62	74
π_{d3}	245.6	1.924	90	90	72	66	68	70
π_{d4}	248.7	1423.8	**100**	**100**	**100**	**100**	**100**	**100**

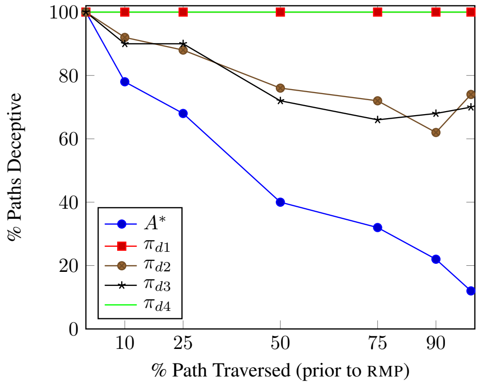

Figure 4: Results show the percentage of paths returned by each strategy that were deceptive when tested at 10%, 25%, etc., of their path length *prior to the* RMP (beyond the RMP, all paths are truthful). Table columns show average (total) path costs and average time taken to generate the (total) path. Generation time for all strategies exceeded that of A* by an order of magnitude. See inline text for discussion of time taken to generate π_{d4}. π_{d1} and π_{d4} were both strongly deceptive to the full extent but π_{d4} achieved this at much lower cost. (Two maps, 50 Moving-AI scenarios, each modified to include three extra goals. Experiments were conducted on a i7 3.6GHz machine with 8GB RAM.)

layout in order to achieve goal recognition more easily, has several interesting parallels with deceptive path-planning. The authors explicitly consider one type of deceptive path that is somewhat similar to our simulation strategy but capped by a budget (as noted in Section 5). The *grd* concept of worst case distinctiveness (*wcd*)—the maximum distance that an agent can travel along any path in the domain without revealing her goal—is superficially similar to our LDP. The *wcd*, however, is a property of the domain (not the path) and is measured by counting steps. When considering suboptimal paths, therefore, (whose lengths are potentially infinite), the authors encounter the problem discussed at 4.1. They impose a budgetary constraint, which they call "bounded non-optimality" [Keren *et al.*, 2015]. Interestingly, modification of a domain using *grd* can make deceptive path-planning easier, as it allows simulation (the unambiguous targeting of a bogus goal) to begin at an earlier point.

The general problem of deception/privacy and location tracking arises in many other settings which we mention here only briefly: amongst the robotics community, for example, in consideration of the panda tracker problem [O'Kane, 2009] and in numerous ambush, pursuit-evasion and patrolling games [Shieh *et al.*, 2012].

7 Conclusion

In this paper, we have presented a model of deceptive path-planning. We have introduced the notion of a last deceptive point and, in identifying its position, have demonstrated—in path completion—a novel method of assessing distance travelled along a suboptimal path. Based on [Masters and Sardina, 2017], we presented a formula to capture a theoretical lower bound for the 'radius of maximum probability' without needing to calculate any actual probabilities. Finally, we have discussed strategies to compute deceptive paths, including one capable of achieving fully deceptive rational paths.

Our work is limited by two simplifying assumptions with respect to the observer: that she is naive (does not expect to be deceived) and rational (inherited from the goal recognition paradigm on which the model of the observer is based). Of course, if an observer were to see the deceiving agent on multiple occasions, one might expect that her goal recognition model would change/adapt, for example if using a machine learning approach (e.g., [Liao *et al.*, 2007]). Nevertheless, for one-time, or first-time, scenarios, we believe these assumptions to be reasonable.

In future work, we will incorporate magnitude into our measurement of deception and explore how the three dimensions can be combined into one index so that the relative deceptivity of different paths can be more easily compared. There is scope to develop optimised algorithms to implement the proposed strategies. We also wish to introduce more sophisticated strategies, including those that recognise known (or suspected) idiosyncrasies in the observer.

Acknowledgments

We thank the reviewers for their suggestions and acknowledge the support of the Australian Government Research Training Program Scholarship and the Australian Research Council (DP120100332).

References

[Alloway *et al.*, 2015] Tracy P. Alloway, Fiona McCallum, Ross G. Alloway, and Elena Hoicka. Liar, liar, working memory on fire: investigating the role of working memory in childhood verbal deception. *Journal of experimental child psychology*, 137:30–38, 2015.

[Arkin *et al.*, 2012] Ronald C. Arkin, Patrick Ulam, and Alan R. Wagner. Moral decision making in autonomous systems: enforcement, moral emotions, dignity, trust, and deception. *Proc. of the Institute of Electrical and Electronics Engineers (IEEE)*, 100(3):571–589, 2012.

[Baker *et al.*, 2009] C. Baker, R. Saxe, and J.B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[Bast *et al.*, 2015] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks. Technical Report MSR-TR-2014-4, Microsoft Corporation, 2015.

[Bell, 2003] J. Bowyer Bell. Toward a theory of deception. *International Journal of Intelligence and Counterintelligence*, 16(2):244–279, 2003.

[Bowyer, 1982] J. Barton Bowyer. *Cheating: Deception in War & Magic, Games & Sports*. St Martin's Press, 1982.

[Charniak and Goldman, 1991] Eugene Charniak and Robert P. Goldman. Probabilistic abduction for plan recognition. Technical report, Brown University and Tulane University, 1991.

[Demolombe and Hamon, 2002] Robert Demolombe and Erwan Hamon. What does it mean that an agent is performing a typical procedure?: a formal definition in the situation calculus. In *Proc. of IJCAI*, pages 905–911, 2002.

[Dias *et al.*, 2013] Joao Dias, Ruth Aylett, Henrique Reis, and Ana Paiva. The great deceivers: Virtual agents and believable lies. *Cognitive Science*, pages 2189–2194, 2013.

[Hart *et al.*, 1968] Peter. E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. 4(2):100–107, 1968.

[Hespanha *et al.*, 2000] João P. Hespanha, Yusuf S. Ateskan, H. Kizilocak, et al. Deception in non-cooperative games with partial information. pages 139–147, 2000.

[Hespanha, 2006] João P. Hespanha. Application and value of deception. *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, pages 145–165, 2006.

[Jian *et al.*, 2006] Jiun-Yin Jian, Toshihiko Matsuka, and Jeffrey V. Nickerson. Recognizing deception in trajectories. In *Proc. of the Cognitive Science Society*, pages 1563–1568, 2006.

[Kautz and Allen, 1986] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *Proc. of AAAI*, pages 32–37, 1986.

[Keren *et al.*, 2014] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design. In *Proc. of ICAPS*, pages 154–162, 2014.

[Keren *et al.*, 2015] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design for non-optimal agents. In *Proc. of AAAI*, pages 3298–3304, 2015.

[Keren *et al.*, 2016] Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy preserving plans in partially observable environments. In *Proc. of IJCAI*, pages 3170–3176, 2016.

[Korf, 1990] Richard E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2):189–211, 1990.

[LaValle, 2006] Stephen M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

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

[Masters and Sardina, 2017] Peta Masters and Sebastian Sardina. Cost-based goal recognition for path-planning. In *Proc. of AAMAS*, pages 750–758, 2017.

[Millington and Funge, 2009] Ian Millington and John Funge. *Artificial Intelligence in Games*. Morgan Kaufmann, Burlington, Massachusetts, 2nd edition, 2009.

[O'Kane, 2009] Jason M. O'Kane. On the value of ignorance: Balancing tracking and privacy using a two-bit sensor. In *Algorithmic Foundation of Robotics VIII*, pages 235–249. 2009.

[Ramirez and Geffner, 2009] Miquel Ramirez and Hector Geffner. Plan recognition as planning. In *Proc. of IJCAI*, pages 1778–1783, 2009.

[Ramirez and Geffner, 2010] Miquel Ramirez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proc. of AAAI*, pages 1121–1126, 2010.

[Root *et al.*, 2005] Philip Root, Jan De Mot, and Eric Feron. Randomized path planning with deceptive strategies. In *Proc. of the American Control Conference*, pages 1551–1556, 2005.

[Sharkey and Sharkey, 2012] Amanda Sharkey and Noel Sharkey. Granny and the robots: ethical issues in robot care for the elderly. *Ethics and Information Technology*, 14(1):27–40, 2012.

[Shieh *et al.*, 2012] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. PROTECT: A deployed game theoretic system to protect the ports of the United States. In *Proc. of AAMAS*, pages 13–20, 2012.

[Shim and Arkin, 2012] Jaeeun Shim and Ronald C. Arkin. Biologically-inspired deceptive behavior for a robot. In *From Animals to Animats 12*, pages 401–411. 2012.

[Shim and Arkin, 2013] Jaeeun Shim and Ronald C. Arkin. A taxonomy of robot deception and its benefits in hri. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2328–2335, 2013.

[Siegwart *et al.*, 2011] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT press, 2011.

[Sturtevant, 2012] Nathan R Sturtevant. Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):144–148, 2012.

[Turing, 1950] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[Whaley, 1982] Barton Whaley. Toward a general theory of deception. *The Journal of Strategic Studies*, 5(1):178–192, 1982.