An Optimization Approach to Privacy-Preserving Path Planning

Paper # 85

Abstract

In this paper, we consider the problem of privacy-preserving path planning in which an actor moves in an environment to reach a destination that it wants to keep private while an observer monitors the actor's actions to discover such a destination. Several real-world applications in surveillance, reconnaissance, intelligent transportation, and computer games, to mention a few, can be modeled within this framework. The problem is an instance of goal obfuscation, which has lately received significant attention in the AI community. With several different variants being proposed, the field lacks coherence and characterization from first principles. Here, we analyze goal obfuscation in the context of path planning, casting it into a formal optimization problem. We propose a complete theoretical analysis in the case of deterministic actions and full observability and introduce efficient algorithms to find exact solutions. Our experiments show that such algorithms scale well to solve complex problems.

Introduction

In this paper, we consider an adversarial path planning setting with two agents: an *actor*, which moves in the environment to reach a destination that it wants to maintain private, and an *observer*, which monitors the actor's actions to discover such a destination. We offer algorithms to assist the actor in finding paths to reach its goals as efficiently as possible while preserving its privacy.

Our setting is relevant because it can be used to model several real-world applications. Many problems in engineering and computer science can be seen as finding a path in a graph and, in many of those contexts, privacy is a crucial factor. In law enforcement, for example, the police aims to escort important individuals to locations that must remain secret; in military settings, troops need to move towards targets without revealing them; and, in computer games, the behavior of the players must be sufficiently unintelligible for each session to be entertaining.

We define *privacy* in terms of how far the actor can proceed towards its destination before revealing it to the observer. A destination is revealed when it becomes the only possible goal compatible with the actor's behavior. In this context, we provide the actor with a fast method to analyze the environment off-line and determine a set of paths to destination (its *strategy*) that preserves its privacy. In particular, we present two techniques: the first allows the actor to select the cheapest strategy that ensures a desired level of privacy; the second enables the actor to select the strategy that maximizes its privacy within its available resource budget.

Our techniques can be seen as incarnations of *goal obfuscation*, which, as defined in Chakraborti et al. (2019), is concerned with 'increasing the ambiguity over the possible goals that might be achieved'. Lately, there has been significant interest in this topic in the AI community as well as in legibility, which is the counterpart of obfuscation (Kulkarni, Srivastava, and Kambhampati 2019; Keren, Gal, and Karpas 2015; Dragan, Lee, and Srinivasa 2013). Both themes are in turn part of a broader area of AI that studies interpretable agent behavior.

The current body of work in goal obfuscation is set out in task-planning settings, mostly with partial observability (Kulkarni, Srivastava, and Kambhampati 2019), and primarily finds solutions within the realms of goal recognition and goal recognition design (Masters and Sardiña 2017; Keren, Gal, and Karpas 2015; Keren, Gal, and Karpas 2016b). Instead, we tackle the problem of goal obfuscation in the context of path planning (with deterministic actions and full observability) from first principles. Path planning is a simplified setting compared to task planning since the actor's and the observer's models are identical. This simplification dispenses us with handling the complexity of possible mismatches between those models and allows us to give a crisp mathematical formulation of goal obfuscation as an optimization problem. We study such a problem and offer efficient algorithms to solve privacy-preserving path planning, which are not currently available in task planning.

Previous work in interpretable agent behavior in the context of path planning is limited to deception (Masters and Sardiña 2017), which is different from goal obfuscation in that deception involves actively misleading the observer, which we do not consider (see Related Work).

Problem Statement

Given a directed graph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ with a source node $o \in \mathcal{V}$ (with only outgoing edges) and a set of destinations nodes

$\mathcal{D} \subseteq \mathcal{V}$ (with only incoming edges), let us consider an *actor* that is initially in o and aims to reach one of the destinations in $\mathcal{D}$ by moving over $\mathcal{H}$. A second agent, the *observer*, can observe the actor at every node and wants to discover where the actor is going. We assume that $\mathcal{H}$, o, and $\mathcal{D}$ are known to both agents.

We call a *path* in $\mathcal{H}$ any sequence of vertices $\gamma = (\gamma_0, \dots, \gamma_l)$ such that each pair $(\gamma_i, \gamma_{i+1}) \in \mathcal{E}$. We say that γ is connecting γ_0 to γ_l and that l is the *length* of γ. Given a, b such that $0 \leq a \leq b \leq l$, we define the subpath $\gamma|_a^b = (\gamma_a, \dots, \gamma_b)$.

Consider now a set $\mathcal{P}$ of paths on $\mathcal{H}$ from o to the destination nodes. We say that $\mathcal{P}$ is $(o, \mathcal{D})$-*connecting* if, for every $d \in \mathcal{D}$, there is at least one path in $\mathcal{P}$ that connects o to d. Notice that, because of the assumptions made on the edges from the origin and into the destinations, paths considered cannot include o or any node in $\mathcal{D}$ as intermediate nodes.

We consider the map $\delta : \mathcal{P} \to \mathcal{D}$ such that $\delta(\gamma)$ is the destination node of γ. We use the symbol $\mathcal{P}^k$ to indicate the set of prefixes of length k of all the paths in $\mathcal{P}$ (having length k). Precisely, if $\gamma = (\gamma_0, \dots, \gamma_l) \in \mathcal{P}$, the prefix of γ of length $k \leq l$ is $\gamma^k = \gamma|_0^k = (\gamma_0, \dots, \gamma_k)$.

A strategy for the actor is to choose a set $\mathcal{P}$ that is $(o, \mathcal{D})$-*connecting*. We associate a cost to the actor's strategy. We model this cost by introducing a non-negative weight matrix W whose rows and columns are labelled by the graph nodes: if $(i, j) \in \mathcal{E}$, W_{ij} can be interpreted as the distance along that edge or the time needed to traverse it. The weight of a path $\gamma = (\gamma_0, \dots, \gamma_l)$ is defined as

$$W(\gamma) = \sum_{h=0}^{l-1} W_{\gamma_h, \gamma_{h+1}}$$

We denote with $\omega(i, j)$ the geodetic distance in $\mathcal{G}$ between the nodes i and j relatively to W (this is the minimum weight of a path connecting i to j in $\mathcal{G}$). Finally, we define the cost of paths as their relative weights with respect to the geodetic distance, namely

$$C(\gamma) = \frac{W(\gamma)}{\omega(o, \delta(\gamma))}, \quad C(\mathcal{P}) = \max_{\gamma \in \mathcal{P}} C(\gamma)$$

Our theory and algorithms remain unchanged if absolute weights are used as a measure of cost.

The actor's performance relates to the position that the actor has achieved along the path towards its destination when the observer discovers it. To formalize this concept, we set some additional notation and concepts.

Definition 1. *Given a set of paths $\mathcal{P}$ from o to $\mathcal{D}$, a path $\gamma \in \mathcal{P}$, and a positive integer t, we say that γ* **discloses** *$\mathcal{D}$ at step t if, for every $\gamma' \in \mathcal{P}$ such that $\gamma^t = \gamma'^t$, it holds that $\delta(\gamma) = \delta(\gamma')$.*

Definition 1 says that if the actor follows the path γ for t steps, the observer can univocally decide which destination the actor is going to because all paths in $\mathcal{P}$ compatible with its observed behavior lead to the same destination.

Definition 2. *Given a set of paths $\mathcal{P}$ from o to $\mathcal{D}$ and a path $\gamma = (\gamma_0, \dots, \gamma_l) \in \mathcal{P}$, we now define the following indices:*

- *The* **disclosing index** *of γ is the minimum t for which γ discloses $\mathcal{D}$ at step t and is denoted by $t_\gamma(\mathcal{P})$. This index gives us the first step at which the intentions of the actor are revealed to the observer.*
- *The* **disclosing distance** *of γ is a measure of how far is the actor from its actual destination when the observer discovers it. Formally, it is expressed as*

$$\lambda_\gamma(\mathcal{P}) = \sum_{t=t_\gamma(\mathcal{P})}^{l-1} W_{\gamma_t, \gamma_{t+1}} \tag{1}$$

- *The* **upper disclosing distance** *of $\mathcal{P}$ is defined as*

$$\lambda(\mathcal{P}) = \max_{\gamma \in \mathcal{P}} \lambda_\gamma(\mathcal{P})$$

The upper disclosing distance is a conservative index, which is useful in "worst case" scenarios where the actor wants to guarantee a certain performance threshold. In particular, minimizing $\lambda(\mathcal{P})$ ensures that the actor's distance to the chosen destination is below a certain threshold when such a destination becomes clear to the observer.

Example 1. *Consider the graph $\mathcal{H}$ depicted in Fig. 1. We assume that all edges (i, j) have weight $W_{ij} = 1$. We call $\mathcal{P}$ the set of all paths in $\mathcal{H}$ from o to the four destinations $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$ that have minimal cost (respectively, costs $2, 4, 3, 4$). Paths costs, disclosing indices and distances are reported in Table 1. Consequently, $\lambda(\mathcal{P}) = 3$.*

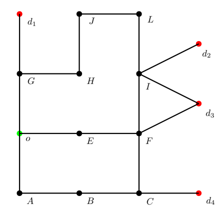

Figure 1: Example of a graph with 4 destinations.

$\gamma \in \mathcal{P}$	$C(\gamma)$	$t_{\gamma,\mathcal{P}}$	$\lambda_{\gamma,\mathcal{P}}$
OGd_1	2	1	1
$OEFId_2$	4	3	1
$OEFd_3$	3	3	0
$OEFCd_4$	4	3	1
$OABCd_4$	4	1	3

Table 1: Costs, disclosing indices and distances for the set of minimal cost paths $\mathcal{P}$ connecting o to $\mathcal{D}$.

We now introduce the two optimization problems that we study in this paper:

- Find a set of paths $\mathcal{P}$ that minimizes the cost $C(\mathcal{P})$ within those that are $(o, \mathcal{D})$-connecting and have the upper disclosing distance below a given threshold λ. Formally,

$$C^{\min}(\lambda) = \min_{\substack{\mathcal{P}\ (o,\mathcal{D})-\text{conn.} \\ \lambda(\mathcal{P}) \leq \lambda}} C(\mathcal{P}) \tag{2}$$

- Find a set of paths that minimizes the upper disclosing distance $\lambda(\mathcal{P})$ within those that are $(o, \mathcal{D})$-connecting and have a cost below a given threshold C. Formally,

$$\lambda^{\min}(C) = \min_{\substack{\mathcal{P}\ (o,\mathcal{D})-\text{conn.} \\ C(\mathcal{P}) \leq C}} \lambda(\mathcal{P}) \tag{3}$$

A set $\mathcal{P}$ that minimizes Eq. (2) represents the cheapest set of paths that allows the actor to maintain its disclosing distance below λ. If the actor cannot afford a cost higher than C, the choice of any set of paths $\mathcal{P}$ that minimizes Eq. (3) ensures that it achieves the best possible upper disclosing distance. If $\lambda^{min}(C) = 0$, goal obfuscation is total: the destination will be disclosed when the actor eventually reaches it. Instead, if $\lambda^{min}(C) > 0$, obfuscation is only partial.

Path Covering Properties

In this section, we undertake an in-depth study of the notion of upper disclosing distance introduced in Definition 2. In particular, we prove a characterization result of fundamental importance in constructing the algorithms to solve the optimization problems in Eqs. (2) and (3).

We start with the following pairwise relation that is at the core of the concept of disclosing distance.

Definition 3. *Given two paths γ_1 and γ_2 from o to $\mathcal{D}$, we say that γ_2* **covers** *γ_1 at level $\lambda \geq 0$ iff:*

- $\delta(\gamma_1) \neq \delta(\gamma_2)$;
- $\lambda_{\gamma_1}(\{\gamma_1, \gamma_2\}) \leq \lambda$

We will use the notation $\gamma_2 \rightarrow_\lambda \gamma_1$ to denote that γ_2 covers γ_1 at level λ.

Definition 3 has the following interpretation: when $\mathcal{P}$ encompasses only the two paths γ_1 and γ_2, if the agent follows path γ_1, it will disclose its destination when it is within distance λ from it. Note that the covering relation naturally leads to a directed graph structure on the set of all paths from o to $\mathcal{D}$, which we denote with the symbol $\mathcal{H}_{\mathcal{P}}(\lambda)$. When both $\gamma_2 \rightarrow_\lambda \gamma_1$ and $\gamma_1 \rightarrow_\lambda \gamma_2$, we write $\gamma_2 \leftrightarrow_\lambda \gamma_1$. The graph $\mathcal{H}_P(\lambda)$ is monotonically increasing in λ.

The following result holds true:

Proposition 4. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths and let $\gamma \in \mathcal{P}$. Then, for every $\lambda \geq 0$,*

$$\exists \gamma' \in \mathcal{P} : \gamma' \rightarrow_\lambda \gamma \quad \Leftrightarrow \quad \lambda_\gamma(\mathcal{P}) \leq \lambda$$

Proof $\Rightarrow$: Let $t \geq 0$ be such that $\gamma^{t-1} = \gamma'^{t-1}$ and $\gamma_t \neq \gamma'_t$. Then, it must hold $W(\gamma|_t^l) \leq \lambda$ where l is the length of γ. Given the definition of disclosing index, we have that $t_\gamma(\mathcal{P}) \geq t$ and thus $\lambda_\gamma(\mathcal{P}) = W(\gamma|_{t_\gamma(\mathcal{P})}^l) \leq \lambda$.

$\Leftarrow$: It can be proven similarly, inverting the steps above. ■

The following result plays a crucial role in the development of our theory and in the design of the optimization algorithms. It says that a set of paths can achieve a certain threshold λ for the upper disclosing distance only if specific covering relations involving either pairs or triples of paths exist among them. More specifically, for every path γ, either γ is part of a pair (γ, γ') such that γ and γ' cover each other or γ is covered by another pair (γ', γ'') such that γ' and γ'' cover each other. Formally,

Theorem 5. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths and let $\lambda \geq \lambda(\mathcal{P})$. Then, for every $\gamma \in \mathcal{P}$, at least one of the following facts must hold true:*

1. *There exists $\gamma' \in \mathcal{P}$ such that $\gamma' \leftrightarrow_\lambda \gamma$;*
2. *There exist $\gamma', \gamma'' \in \mathcal{P}$ such that $\gamma'' \leftrightarrow_\lambda \gamma' \rightarrow_\lambda \gamma$.*

Proof The proof of this result is presented in the Appendix at the end of the paper. ■

We now introduce some notation. Given destinations $d, d_1, d_2 \in \mathcal{D}$ and $\lambda \geq 0$, we define the following subsets:

- $N_d^\lambda = \{i \in \mathcal{V} \,|\, \omega(i, d) \leq \lambda\}$, the *neighborhood* of d consisting of all the nodes within distance λ from d;
- $\partial N_d^\lambda = \{i \in (\mathcal{V} \setminus N_d^\lambda) \,|\, \exists j \in N_d^\lambda \text{ s.t. } (i, j) \in \mathcal{E}\}$, the *boundary* of neighborhood N_d^λ: it consists of the nodes having an edge leading into N_d^λ;
- $\overline{N}_d^\lambda = N_d^\lambda \cup \partial N_d^\lambda$, the *closed neighborhood* of d obtained by unioning N_d^λ and its boundary;
- $\overline{N}_{d_1,d_2}^\lambda = \overline{N}_{d_1}^\lambda \cap \overline{N}_{d_2}^\lambda, \quad N^{2,\lambda} = \bigcup_{(d_1,d_2) \in \mathcal{D} \times \mathcal{D}} \overline{N}_{d_1,d_2}^\lambda$;
- $\widetilde{\partial N}_d^\lambda = \partial N_d^\lambda \setminus N^{2,\lambda}$ with the convention that $\widetilde{\partial N}_d^\lambda = \{o\}$ if the right hand side set is empty. It is the boundary of neighborhood N_d^λ after removing the nodes in $N^{2,\lambda}$.

We now illustrate these objects in a simple example.

Example 2. *In Figure 2, we consider a 10×10 grid graph with unitary weights (we do not report edges among nodes for better visibility). We assume that there are 4 destinations: $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$ and consider $\lambda = 1$. The shaded areas represent the four closed neighbors $\overline{N}_{d_i}^1$ for $i = 1, \ldots, 4$. In darker color, the intersection $\overline{N}_{d_2,d_3}^1$ is depicted. The parts of the closed neighborhoods of d_1 and d_4 within the dashed perimeters are the open neighborhoods, respectively, $N_{d_1}^1$ and $N_{d_4}^1$, while the parts outside the dashed perimeters are the boundaries, $\partial N_{d_1}^1$ and $\partial N_{d_4}^1$, respectively.*

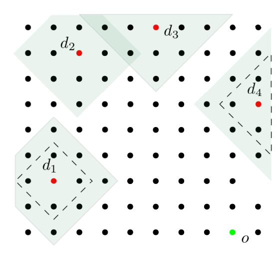

Figure 2: Graphical representation of neighborhoods and boundaries for a grid 10×10 and destinations d_1, d_2, d_3, d_4.

Definition 6. *Two distinct destinations $d_1 \neq d_2$ are called λ-***interacting** *if $\overline{N}_{d_1,d_2}^\lambda \neq \emptyset$.*

We further define:

$$\mathcal{D}_\lambda^2 := \{(d_1, d_2) \in \mathcal{D} \times \mathcal{D} \,|\, d_1,\, d_2\ \lambda\text{-interacting}\}$$

$$\mathcal{D}_\lambda^1 := \{d \in \mathcal{D}, |\, \exists d' \in \mathcal{D} \,|\, :\, (d,\, d') \in \mathcal{D}_\lambda^2\}$$

$$\lambda^* = \min\{\lambda \geq 0 \,|\, \mathcal{D}_\lambda^2 \neq \emptyset\}$$

Example 3. *With reference to Figure 2, notice that d_2 and d_3 are 1-interacting and no other pairs in the graph is 1-interacting. Therefore, in this case,*

$$\mathcal{D}_1^2 = \{(d_2, d_3),\, (d_3, d_2)\}, \quad \mathcal{D}_\lambda^1 = \{d_2,\, d_3\}$$

Since d_2 and d_3 are not 0-interacting, it follows that $\mathcal{D}_0^2 = \varnothing$ and thus $\lambda^* = 1$.

As a first application of Theorem 5, we obtain the following fundamental limitation on the disclosing distance that actor can achieve, regardless of the budget at its disposal.

Corollary 7. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths. Then,*

$$\lambda(\mathcal{P}) \geq \lambda^*$$

Proof Note that if γ' and γ'' are two paths in $\mathcal{P}$ of length, respectively, l' and l'' and such that $\gamma' \leftrightarrow_\lambda \gamma''$, then, necessarily, there exists $h \leq l', l''$ such that $\gamma'|_0^h = \gamma''|_0^h$ and $W(\gamma'|_{h+1}^{l'}), W(\gamma''|_{h+1}^{l''}) \leq \lambda$. This implies that $\gamma'_h = \gamma''_h \in \overline{N}^\lambda_{\delta(\gamma')} \cap \overline{N}^\lambda_{\delta(\gamma'')}$. Therefore, $\mathcal{D}^2_\lambda \neq \varnothing$ and, given the definition of λ^*, it follows that $\lambda \geq \lambda^*$. If we fix $\lambda = \lambda(\mathcal{P})$, the existence of at least a pair of paths such as γ' and γ'' is guaranteed by Lemma 5. ■

Remark 8. In the special case where all weights are unitary (e.g. $W_{ij} \in \{0, 1\}$ for every pair of nodes i and j), then,

$$\lambda^* = \lceil \omega^{\min}/2 \rceil - 1$$

where

$$\omega^{\min} = \min_{\substack{d_1, d_2 \in \mathcal{D} \\ d_1 \neq d_2}} \omega(d_1, d_2)$$

The construction proposed in the next section will show that the inequality in Corollary 7 is sharp: the limit λ^* can always be achieved if a sufficient budget is available.

Algorithms for Optimization

In this section, we propose an exact algorithm for the solution of the optimization problems in Eqs. (2) and (3). The first step is the construction of a family of small sets of paths that can achieve any disclosing distance $\lambda \geq \lambda^*$.

A Canonical Construction

For every two nodes $i, j \in \mathcal{V}$, we take $\gamma^{i,j}$, a path in $\mathcal{G}$ from i to j of minimal weight ($W(\gamma^{i,j}) = \omega(i,j)$). For nodes $i_1, i_2, \ldots, i_s \in \mathcal{V}$, we define

$$\gamma^{i_1,\ldots,i_s} = \gamma^{i_1,i_2} \perp \cdots \perp \gamma^{i_{s-1},i_s}$$

where $\perp$ indicates path concatenation.

We now fix $\lambda \geq 0$ and take the following families of paths:

- For every $(d, d') \in \mathcal{D}^2_\lambda$ and $h \in \overline{N}^\lambda_{d,d'}$, define $\mathcal{P}^{(h)}_{d,d'} = \{\gamma^{o,h,d},\ \gamma^{o,h,d'}\}$. Choose any $h^* \in \overline{N}^\lambda_{d,d'}$ for which $C(\mathcal{P}^{(h)}_{d,d'})$ is minimal and put $\mathcal{P}^\lambda_{d,d'} = \mathcal{P}^{(h^*)}_{d,d'}$.
- For every $d \in \mathcal{D}$ and pair of destinations $(d', d'') \in \mathcal{D}^2_\lambda$ with $d \notin \{d', d''\}$, for every $h_1 \in \widetilde{\partial N}^\lambda_d$ and $h_2 \in \overline{N}^\lambda_{d',d''}$, put $\mathcal{P}^{(h_1,h_2)}_{d,d',d''} = \{\gamma^{o,h_1,d}, \gamma^{o,h_1,h_2,d'}, \gamma^{o,h_1,h_2,d''}\}$. Choose now any $(d'^*, d''^*) \in \mathcal{D}^2_\lambda$, $h_1^* \in \widetilde{\partial N}^\lambda_d$ and $h_2^* \in \overline{N}^\lambda_{d',d''}$ for which $C(\mathcal{P}^{(h_1,h_2)}_{d,d',d''})$ is minimal and put $\mathcal{P}^\lambda_d = \mathcal{P}^{(h_1^*,h_2^*)}_{d,d'^*,d''^*}$.

We emphasize that the construction of the sets $\mathcal{P}^\lambda_{d,d'}$ and $\mathcal{P}^\lambda_d$ is in general not unique as it depends on the selection of minimal paths and on other optimal (typically not unique) choices. This aspect does not impact our future analysis.

Finally, we define:

$$\mathcal{P}^\lambda = \bigcup_{(d,d') \in \mathcal{D}^2_\lambda} \mathcal{P}^\lambda_{d,d'} \cup \bigcup_{d \in \mathcal{D}} \mathcal{P}^\lambda_d \tag{4}$$

Example 4. *In Figure 3, we show two examples of our canonical construction. In particular, in the left part, we represent the optimal pair of paths $\mathcal{P}^1_{d_2,d_3} = \{\gamma^{o,h^*,d_2},\ \gamma^{o,h^*,d_3}\}$ with the bifurcation node $h^* \in \overline{N}^1_{d_2,d_3}$. In the right part, we represent the optimal triple of paths $\mathcal{P}^1_{d_4} = \mathcal{P}^{(h_1^*,h_2^*)}_{d_4,d_2,d_3}$ with the two bifurcation nodes $h_1^* \in \partial N^1_{d_4}$ (in this case, this coincides with $\widetilde{\partial N}^1_{d_4}$) and $h_2^* \in \overline{N}^1_{d_2,d_3}$. Finally, Figure 4 represent the complete set of paths $\mathcal{P}^1$.*

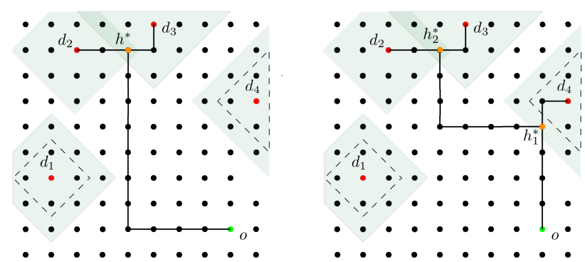

Figure 3: Two examples of canonical constructions.

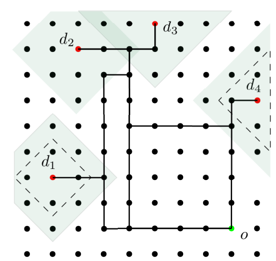

Figure 4: Complete set of paths $\mathcal{P}^1$.

The following result illustrates the first basic properties of the set $\mathcal{P}^\lambda$ that we have introduced.

Theorem 9. *Let $\lambda \geq \lambda^*$. Then, $\mathcal{P}^\lambda$ is $(o, \mathcal{D})$-connecting and $\lambda(\mathcal{P}^\lambda) \leq \lambda$.*

Proof Note first that, by definition of λ^*, we have that $\mathcal{D}^2_\lambda$ is not empty. This implies that, for any destination $d \in \mathcal{D}$, by the way $\mathcal{P}^\lambda_d$ is defined, it holds that $\mathcal{P}^\lambda_d \neq \varnothing$. This proves that $\mathcal{P}^\lambda$ is $(o, \mathcal{D})$-connecting. Finally, the fact that $\lambda(\mathcal{P}^\lambda) \leq \lambda$ follows from Proposition 4. ■

We now present the main result of this section, which shows the universal role of the constructed set of paths $\mathcal{P}^\lambda$. The theorem asserts that, if a certain threshold λ is achievable for the upper disclosing distance given a set of paths whose cost is below a certain budget C, then the same performance is achievable by a suitable subset $\mathcal{P}^\lambda$. This will imply that, when searching for a set of paths of minimal cost achieving a given threshold λ, we can restrict our search within this $\mathcal{P}^{\lambda}$.

Theorem 10. *Given $\lambda \geq 0$, for every set of paths $\mathcal{P}$ that is $(o, \mathcal{D})$-connecting and such that $\lambda(\mathcal{P}) \leq \lambda$, there exists $\mathcal{Q} \subseteq \mathcal{P}^{\lambda}$ that is $(o, \mathcal{D})$-connecting and such that:*

$$\lambda(\mathcal{Q}) \leq \lambda, \quad C(\mathcal{Q}) \leq C(\mathcal{P})$$

Proof We construct $\mathcal{Q}$ in the following way. For every $\gamma \in \mathcal{P}$, we determine a corresponding set of paths $\mathcal{Q}_{\gamma} \subseteq \mathcal{P}^{\lambda}$, of cardinality 2 or 3, containing in particular a path $\tilde{\gamma}$ such that $\delta(\gamma) = \delta(\tilde{\gamma})$. Moreover, $\mathcal{Q}_{\gamma}$ satisfies the following two properties:

1. $\lambda_{\sigma}(\mathcal{Q}_{\gamma}) \leq \lambda$ for every $\sigma \in \mathcal{Q}_{\gamma}$;
2. $C(Q_{\gamma}) \leq C(\mathcal{P})$.

We take $\mathcal{Q}$ as the union of these sets $\mathcal{Q}_{\gamma}$. By construction, $\mathcal{Q} \subseteq \mathcal{P}^{\lambda}$ is $(o, \mathcal{D})$-connecting, and, by property 2., $C(\mathcal{Q}) \leq C(\mathcal{P})$. Moreover, because of property 1., for each $\sigma \in \mathcal{Q}_{\gamma}$, there exists $\sigma' \in \mathcal{Q}_{\gamma}$ such that $\sigma' \to_{\lambda} \sigma$. This implies that $\lambda_{\sigma}(\mathcal{Q}) \leq \lambda$ and shows that $\mathcal{Q}$ has the desired properties. We are thus left with proving the existence of the sets $\mathcal{Q}_{\gamma}$ that satisfy the above properties 1. and 2.. This is done by means of Lemma 5.

Suppose $\gamma \in \mathcal{P}(d)$ satisfies condition 1. of Lemma 5, namely there exists $\gamma' \in \mathcal{P}(d')$ such that $\gamma' \leftrightarrow_{\lambda} \gamma$. In this case, we pick $\mathcal{Q}_{\gamma} = \mathcal{P}^{\lambda}_{d,d'}$. Property 1. is automatically verified. To prove property 2. we proceed as follows. Let t be such that $\gamma'^{t} = \gamma^{t}$ and $\gamma'_{t+1} \neq \gamma_{t+1}$. Necessarily, $h = \gamma'_t = \gamma_t \in \bar{N}^{\lambda}_{d,d'}$. By the definition of $P^{\lambda}_{d,d'}$ and the fact that $C(\gamma^{o,h,d}) \leq C(\gamma)$ and $C(\gamma^{o,h,d'}) \leq C(\gamma')$, it follows that $C(P^{\lambda}_{d,d'}) \leq C(\{\gamma^{o,h,d}, \gamma^{o,h,d'}\}) \leq C(\gamma, \gamma')\} \leq C(\mathcal{P})$.

Suppose now that condition 1. of Lemma 5 does not hold for γ. Then, condition 2. necessarily must hold. Hence, there exist $\gamma' \in \mathcal{P}(d')$ and $\gamma'' \in \mathcal{P}(d'')$ such that $\gamma'' -_{\lambda} \gamma' \to_{\lambda} \gamma$.

If $d \notin \{d', d''\}$, we set $\mathcal{Q}_{\gamma} = \mathcal{P}^{\lambda}_{d}$, otherwise we set $\mathcal{Q}_{\gamma} = \mathcal{P}^{\lambda}_{d',d''}$. Property 1. is automatically verified in both cases.

To prove property 2., we argue as follows. Let $t < t'$ be such that $\gamma'^{t} = \gamma^{t}$ and $\gamma'_{t+1} \neq \gamma_{t+1}$ and $h' = \gamma''^{t'} = \gamma'^{t'}$ and $\gamma''_{t'+1} \neq \gamma'_{t'+1}$. Necessarily, $h = \gamma'_t = \gamma_t \in \overline{N}^{\lambda}_{d}$ and $h' \in \overline{N}^{\lambda}_{d',d''}$. There must exist $s \leq t$ such that $\tilde{h} = \gamma'_s = \gamma_s \in \partial N^{\lambda}_{d}$. Note that $\tilde{h} \notin N^{2,\lambda}$, otherwise condition 1. of Lemma 5 would instead hold true. If $d \notin \{d', d''\}$, by the definition of P^{λ}_{d} and the fact that $C(\gamma^{o,h,d}) \leq C(\gamma)$, $C(\gamma^{o,\tilde{h},h',d'}) \leq C(\gamma')$,and $C(\gamma^{o,\tilde{h},h',d''}) \leq C(\gamma'')$, it follows that

$$\begin{aligned} &C(\mathcal{P}^{\lambda}_{d}) \leq C(\{\gamma^{o,h,d}, \gamma^{o,\tilde{h},h',d'}, \gamma^{o,\tilde{h},h',d''}\}) \\ &\leq C(\gamma, \gamma', \gamma'')\} \leq C(\mathcal{P}) \end{aligned}$$

If $d \in \{d', d''\}$, we repeat the same line of reasoning considering now the definition of $\mathcal{P}^{\lambda}_{d',d''}$ and the fact that $C(\gamma^{o,h',d'}) \leq C(\gamma^{o,\tilde{h},h',d'})$ and $C(\gamma^{o,h',d''}) \leq C(\gamma^{o,\tilde{h},h',d''})$. Proof is now complete. ■

Theorem 10 implies that, when searching for a minimum of the problem in Eq. (2), we can restrict to subsets of $\mathcal{P}^{\lambda}$. Indeed, we have that:

Corollary 11. *Given $\lambda \geq 0$, there exists $\mathcal{Q} \subseteq \mathcal{P}^{\lambda}$ that is $(o, \mathcal{D})$-connecting, $\lambda(\mathcal{Q}) \leq \lambda$ and*

$$C(\mathcal{Q}) = \min_{\substack{\mathcal{P}\ (o,\mathcal{D})-conn. \\ \lambda(\mathcal{P}) \leq \lambda}} C(P)$$

Proof Let $\mathcal{P}$ be any set that achieves the minimum in Eq. (2). By applying Theorem 10, we find $\mathcal{Q} \subseteq \mathcal{P}^{\lambda}$, $(o, \mathcal{D})$-connecting and such that $\lambda(\mathcal{Q}) \leq \lambda$ and $C(\mathcal{Q}) \leq C(\mathcal{P})$. $\mathcal{Q}$ also achieves the minimum in Eq. (2). ■

Algorithms to Compute $C^{\min}(\lambda)$ and $\lambda^{\min}(C)$

A minimal-cost $(o, \mathcal{D})$-connecting subset of $\mathcal{P}^{\lambda}$ can easily be determined in the following way. For every $d \in \mathcal{D}$, first define C^{pair}_{d} to be the minimal cost achievable among the sets $\mathcal{P}^{\lambda}_{d,d'}$ considering all λ-interacting pairs to which d possibly belongs:

$$C^{\text{pair}}_{d} = \min_{\substack{d' \in \mathcal{D}: \\ (d,d') \in \mathcal{D}^2_{\lambda}}} C(\mathcal{P}^{\lambda}_{d,d'}) \tag{5}$$

We use the convention that, if the set over which we calculate the minimum is empty, then we put $C^{\text{pair}}_{d} = +\infty$. This happens if $d \notin \mathcal{D}^1_{\lambda}$.

For every destination, we then define:

$$\mathcal{P}^{\lambda\,\text{opt}}_{d} = \begin{cases} \mathcal{P}^{\lambda}_{d,d'} & \text{if } C^{\text{pair}}_{d} \leq C(\mathcal{P}^{\lambda}_{d}),\ C(\mathcal{P}^{\lambda}_{d,d'}) = C^{\text{pair}}_{d} \\ \mathcal{P}^{\lambda}_{d} & \text{if } C^{\text{pair}}_{d} > C(\mathcal{P}^{\lambda}_{d}) \end{cases} \tag{6}$$

$\mathcal{P}^{\lambda\,\text{opt}}_{d}$ is the set of paths in $\mathcal{P}^{\lambda}$ (either pairs or triples) that guarantees the connection of o to d, maintains the disclosing distance below λ and achieves the minimal possible cost.

Finally, we put:

$$\mathcal{P}^{\lambda\,\text{opt}} = \bigcup_{d \in \mathcal{D}} \mathcal{P}^{\lambda\,\text{opt}}_{d} \tag{7}$$

This is a set of paths that solves the optimization problem in Eq. (2):

Corollary 12. $C(\mathcal{P}^{\lambda\,\text{opt}}) = C^{min}(\lambda)$

Proof Note that, by construction, $\mathcal{P}^{\lambda\,\text{opt}}$ is $(o, \mathcal{D})$-connecting and $\lambda(\mathcal{P}^{\lambda\,\text{opt}}) \leq \lambda$. Consider now any subset $\mathcal{Q} \subseteq \mathcal{P}^{\lambda}$ that is $(o, \mathcal{D})$-connecting and satisfies $\lambda(\mathcal{Q}) \leq \lambda$. We show that $C(\mathcal{P}^{\lambda\,\text{opt}}) \leq C(\mathcal{Q})$. Fix $d \in \mathcal{D}$ and consider any $\gamma \in \mathcal{Q}$ such that $\delta(\gamma) = d$. If there exists $\gamma' \in \mathcal{Q}$ with $\delta(\gamma') = d'$ such that $\gamma' \leftrightarrow_{\lambda} \gamma$, then, given the definition of $\mathcal{P}^{\lambda}_{d,d'}$, we have that:

$$C(\mathcal{P}^{\lambda\,\text{opt}}_{d}) \leq C^{\text{pair}}_{d} \leq C(\mathcal{P}^{\lambda}_{d,d'}) \leq C(\{\gamma, \gamma'\}) \leq C(\mathcal{Q})$$

If, instead, such a $\gamma' \in \mathcal{Q}$ does not exist, there must exist, in virtue of Theorem 5, two paths $\gamma', \gamma'' \in \mathcal{Q}$ such that $\gamma'' \leftrightarrow_{\lambda} \gamma' \to_{\lambda} \gamma$. Given how $\mathcal{P}^{\lambda}_{d}$ has been defined, it holds that:

$$C(\mathcal{P}^{\lambda\,\text{opt}}_{d}) \leq C(\mathcal{P}^{\lambda}_{d}) \leq C(\{\gamma, \gamma'\, \gamma''\} \leq C(\mathcal{Q})$$

We have thus proven that $C(\mathcal{P}^{\lambda\,\text{opt}}) \leq C(\mathcal{Q})$ for every $\mathcal{Q} \subseteq \mathcal{P}^{\lambda}$ that is $(o, \mathcal{D})$-connecting and satisfies $\lambda(\mathcal{Q}) \leq \lambda$. By Corollary 11, the proof is complete. ■

We now make some comments on this construction.

1. Notice first that, when there is only one λ-interacting pair, $\mathcal{P}^{\lambda\,\mathrm{opt}}$ coincides with $\mathcal{P}^{\lambda}$. For the 10×10 grid analyzed in Example 2, the set of paths represented in Figure 4 is therefore, in that case, the optimal $\mathcal{P}^{1\,\mathrm{opt}}$.
2. In the general case, the cardinality of $\mathcal{P}^{\lambda\,\mathrm{opt}}$ satisfies the bound $|\mathcal{P}^{\lambda\,\mathrm{opt}}| \leq 3|\mathcal{D}|$.
3. Even when λ is so large that all destinations belong to a λ-interacting pair, namely $\mathcal{D} = \mathcal{D}^1_\lambda$, it is not sufficient to only use the pair-type subsets $\mathcal{P}^{\lambda}_{d,d'}$ for the construction of the optimal $\mathcal{P}^{\lambda\,\mathrm{opt}}$. An example is reported in Figure 5. The graph is the grid graph constrained outside of the gray, unpassable areas. On the left, we represent $\mathcal{P}^2_{d_3,d_4}$. This choice is not optimal either for d_3 or for d_4. On the right, we represent the optimal choices $\mathcal{P}^2_{d_3}$ and $\mathcal{P}^2_{d_4}$.

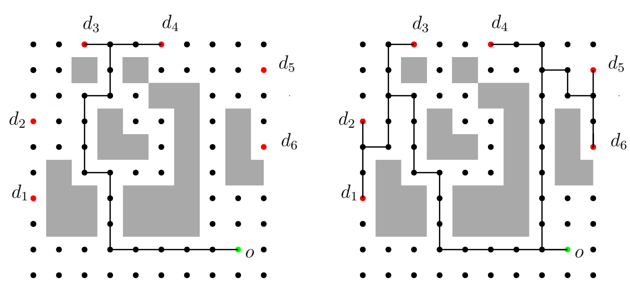

Figure 5: Left: Non-optimal choice for d_3 and for d_4. Right: Optimal choices $\mathcal{P}^2_{d_3}$ and $\mathcal{P}^2_{d_4}$.

In a practical implementation of this algorithm, the construction of the optimal set of paths $\mathcal{P}^{\lambda\,\mathrm{opt}}$ can be done directly, without constructing $\mathcal{P}^{\lambda}$ first. This is illustrated in the pseudo-code reported in Algorithm 1.

The algorithm works in three steps:

1. For destinations d such that $\omega(o,d) \leq \lambda$, it puts $\mathcal{P}^{\lambda\,\mathrm{opt}}_d = \{\gamma^{o,d}\}$ without explicitly constructing a pair or triple covering d, as any other path to another destination will do it (lines 3 to 6);
2. For destinations $d \in \mathcal{D}^1_\lambda$, it constructs the minimum-cost pair covering $\mathcal{P}^{\lambda}_{d,d'}$ (lines 9 to 16);
3. For destinations $d \in \mathcal{D}$ (including those already considered in 2.), it constructs the minimum-cost pair or triple covering $\mathcal{P}^{\lambda\,\mathrm{opt}}_d$ (lines 17 to 25).

The computation complexity of this algorithm is dominated by the computations of the minimal paths between the origin and the bifurcation points, between the different bifurcation points, and between the bifurcation points and the destinations. The number of such computations is upper bounded by $|\mathcal{D}|^3 f(\lambda)$ where:

$$f(\lambda) = 5 \max_{d \in \mathcal{D}} \left[|\partial N^{\lambda}_d| \cdot |\bar{N}^{\lambda}_d| \right]$$

This is obtained considering the worst scenario where, to construct $\mathcal{P}^{\lambda\,\mathrm{opt}}_d$, all possible pairs $\mathcal{P}^{(h)}_{d,d'}$ (with $d' \neq d$) and

Algorithm 1: Optimal Calculation of $\mathcal{P}^{\lambda\,\mathrm{opt}}$

Input: $\mathcal{V}, \mathcal{D}, o, \lambda$
Output: $\mathcal{P}^{\lambda\,\mathrm{opt}}$
1 $BestPaths = \varnothing$;
2 $\mathcal{P}^{\lambda\,\mathrm{opt}} = \varnothing$;
3 **foreach** $d \in \mathcal{D}$ **do**
4   **if** $\omega(o,d) \leq \lambda$ **then**
5   Add $\gamma^{o,d}$ to $\mathcal{P}^{\lambda}$;
6         $BestCost(d) = 1.0$;
7   **else**
8         $BestCost(d) = \infty$;
9 **foreach** $d \in \mathcal{D}$ **do**
10   **if** $\omega(o,d) \leq \lambda$ **then**
11   **continue**;
12   **foreach** $d' \in \mathcal{D}^1_\lambda$ *s.t.* $d \neq d'$ **do**
13   **foreach** $h \in \overline{N}^{\lambda}_{d,d'}$ **do**
14   **if** $BestCost(d) > C(\mathcal{P}^{(h)}_{d,d'})$ **then**
15                 $BestCost(d) = C(\mathcal{P}^{(h)}_{d,d'})$;
16                 $BestPaths(d) = \forall \gamma \in \mathcal{P}^{(h)}_{d,d'}$;
17 **foreach** $d \in \mathcal{D}$ **do**
18   **if** $\omega(o,d) \leq \lambda$ **then**
19   **continue**;
20   **foreach** $(d', d'') \in \mathcal{D}^2_\lambda$ *s.t.* $d \neq d', d''$ **do**
21   **foreach** $h_1 \in \widetilde{\partial N}^{\lambda}_d$ **do**
22   **foreach** $h_2 \in \overline{N}^{\lambda}_{d',d''}$ **do**
23   **if** $BestCost(d) > C(\mathcal{P}^{(h_1,h_2)}_{d,d',d''})$ **then**
24                     $BestCost(d) = C(\mathcal{P}^{(h_1,h_2)}_{d,d',d''})$;
25                     $BestPaths(d) = \forall \gamma \in \mathcal{P}^{(h_1,h_2)}_{d,d',d''}$;
26 **foreach** $\gamma \in BestPaths$ **do**
27   **if** $\gamma \notin \mathcal{P}^{\lambda\,\mathrm{opt}}$ **then**
28   Add γ to $\mathcal{P}^{\lambda\,\mathrm{opt}}$;
29 **return** $\mathcal{P}^{\lambda\mathrm{opt}}$;

triples $\mathcal{P}^{(h_1,h_2)}_{d,d',d''}$ (with $d', d'' \neq d$) need to be computed, considering that the number of minimal paths in each of them is at most 5.

If we call $g(n,m)$ the worst complexity of a minimal path search in a graph of n nodes and m edges, we can conclude that the complexity of our algorithm is bounded by:

$$|\mathcal{D}|^3 f(\lambda) g(n,m)$$

Note that the size of the graph only appears in the term $g(n,m)$: if an optimized implementation of the Dijkstra algorithm is employed for the minimal-path search (as we did in our experiments), we have $g(n,m) = O(m + n \ln n)$. The other two key parameters are $|\mathcal{D}|$, the number of destinations, and λ, the upper disclosing distance threshold. With respect to $|\mathcal{D}|$, the complexity is always cubic, while, with respect to λ, it is determined by the expansivity properties of the graph at hand. For subgraphs of the grids, it holds $f(\lambda) = O(\lambda^3)$.

We now briefly tackle the optimization problem dual to the one of Eq. 2 considered so far, which is given in Eq. (3) and that can be reformulated as follows:

$$\lambda^{\min}(C) = \min\{\lambda \,|\, C^{\min}(\lambda) \leq C\}$$

Although there are direct methods to address this problem, we solve it here by exploiting the solution found for Eq. 2. In particular, our goal is to produce a graph of cost C versus $\lambda^{min}(C)$, which can be consulted to find the best possible λ that can be achieved given a cost. We proceed as follows. We start from $\lambda = \lambda^*$, compute $C_{max} = C^{\min}(\lambda^*)$ and put $\lambda^{\min}(C_{max}) = \lambda^*$. We then take the next $\lambda > \lambda^*$ such that $C = C^{\min}(\lambda) < C_{max}$. We put $\lambda^{\min}(C) = \lambda$. We iterate in such a way until the cost becomes equal to 1 or λ hits the higher value calculated in solving Eq. 2. We obtain a step function for $\lambda^{\min}(C)$, where each cost interval corresponds to the smallest λ that can be achieved by incurring that cost.

Experiments

We performed a large set of experiments. A first batch of experiments was conducted on $N \times N$ square grid graphs (with N equal to 50, 100, 150 and 200) with each node having degree 4, expect for the boundaries nodes having degrees 3 or 2. For each N, we randomly selected origin and destinations (with $|D|$ equal to 4, 6, 8, 10, 12). For each combination of N and D, we run 100 experiments. A second batch was performed on city maps from the 'Moving-AI' 2D pathfinding benchmarks (Sturtevant 2012). The maps are digitalizations of fragments of the cities of Shanghai, Boston, New York and Denver and are represented as 256×256 square grid graphs with unpassable areas (obstacles) and each node having degree 8, except for the boundaries nodes having degrees 5 or 3. As for the first batch, for each city, we performed 100 experiments for $4, 6, 8, 10$ and 12 random destinations. In all cases, the edges have unit cost. We used a server with 8 Intel E5-2583 cores running at 2.10 GHz to perform the experiments. The memory limit by process is 8 GBs.

For both groups of problems, we performed three series of experiments. The first series (Figure 6) focuses on how the runtime (in milliseconds) of the Algorithm 1 changes as a function of the graph size (for grids) or configuration (for cities) and the number of destinations. Runtime for each setting has been averaged over 100 runs. Both figures show that our algorithm is very fast even on very large problems, with a maximum average runtime below 4 minutes for 200×200 square grids and 12 destinations and around 6 minutes for Denver with 12 destinations.

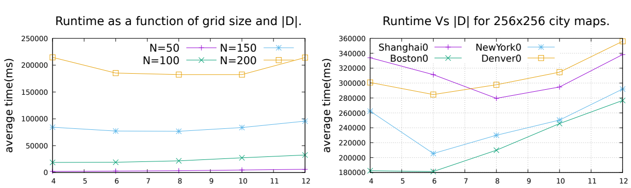

Figure 6: Runtime of Algorithm 1 for grid graphs (left) and city fragments from 'Moving-AI' benchmarks.

The second series of experiments studies how the cost $C^{\min}(\lambda)$ changes when we vary the upper disclosing distance λ. As an example, Figure 7[Left] shows results for an instance of a grid 100×100 with 8 destinations, and Figure 7[Right] shows results for an instance of Shanghai with 12 destinations. The figures illustrate that, typically, when we increase λ enough, we can observe a dramatic drop in the cost. This happens when new bifurcations, cheaper to reach, can be used to cover the different destinations.

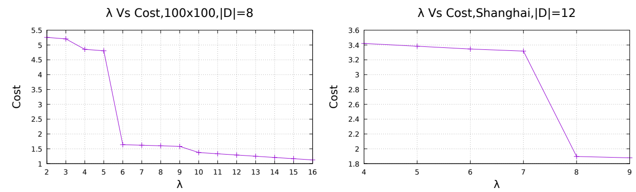

Figure 7: Cost $C^{\min}(\lambda)$ for various values of $\lambda \geq \lambda^*$.

Finally, the third series of experiments looks at how the algorithm scales with increasing λ. Figure 8[Left] shows results for an instance of a grid 100×100 with 8 destinations, and Figure 8[Right] shows results for an instance of Shanghai with 12 destinations. It is apparent that the algorithm scales well in λ as it runs in around 50 seconds for the biggest λ for the grid and around 4.6 minutes for Shanghai.

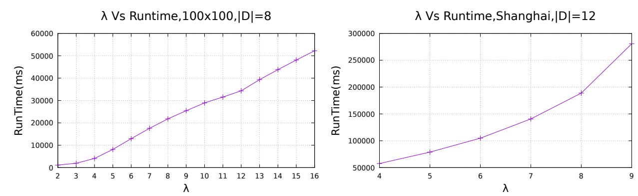

Figure 8: Change in the algorithm's runtime when we vary the upper disclosing distance λ.

Related Work

Our interest is in a rigorous, in-depth study of goal obfuscation in the realm of path planning (PP). PP is the problem of finding a path from a given initial location to a given destination in a map of an environment. PP is one of the most traditional problems in robotics and AI and has been studied extensively (Hart, Nilsson, and Raphael 1968; Harabor and Grastien 2012; LaValle 2006; Cherkassky, Goldberg, and Radzik 1996).

As we are not interested in PP per se, we focus here on the intersection between path planning and goal recognition. To our knowledge, the only previous work in this specific area is *deceptive path planning* (DPP) by Masters and Sardiña (2017). There are several differences between our work and DPP. The definition of the problem is in itself different. While we retain the traditional PP setting, with the modification that we consider a set of possible destinations instead of just one, Masters and Sardiña considers PP within a probabilistic setting and add a set of bogus destinations to the real one. They present two types of strategies, simulation and dissimulation. Simulation corresponds to 'showing the false' and simply accounts for reaching a bogus destination before the real one. The authors note that this strategy is expensive and not effective as soon as the fake destination has been reached. Dissimulation, on the other hand, involves 'hiding the true'. The authors give three ad hoc, non-optimized strategies that all revolve around proceeding towards a point for which the probabilities to go to the real destination and a bogus one are equal, before switching to the real destination. As we do not distinguish between real and fake destinations, our goal is neither simulation or dissimulation. Give a budget, we provide the actor with a portfolio of paths that allows it to proceed as close as possible to any of the possible destinations without disclosing its intentions. Note that, to run our algorithm, we do need to know the actual destination that the actor intends to reach. In this regard, also note that our approach is *secure*, as defined by Kulkarni et al. 2018. Our algorithms return the same output regardless of which destination in the set of possible destinations the actor is targeting, which means that rerunning the algorithms to break privacy is of no use for the observer.

The majority of the work in goal obfuscation has been conducted within the setting of task planning, especially with partial observability. Kulkarni et al. (2019), for example, present a unifying framework for planning with partial information in adversarial and cooperative environments and offer algorithms for computing obfuscated plans as well as legible plans. Reasoning about the behavior of an actor that is interested in maintaining its privacy also plays an important part in two emerging sub-fields of task planning: goal recognition (GR) (Kautz and Allen 1986; Ramirez and Geffner 2009; Pattison and Long 2011) and goal recognition design (GRD) (Keren, Gal, and Karpas 2014; Keren, Gal, and Karpas 2016a). GR takes the observer's point of view and provides algorithms to identify the actor's goal by observing its actions. To come up with an intelligent strategy for the observer, GR might need to reason about the possible strategies of the actor and how to respond to them. GRD, on the other hand, deals with modifying the environment in which the actor lives to facilitate or impede goal recognition. Keren et al. (2015, 2016), in particular, deal with obfuscation in adversarial settings from the goal recognition design aspect.

Conclusions and Future Work

In this paper, we present a mathematical framework for an adversarial path-planning setting where an actor that moves along the nodes of a graph to reach a destination tries to obfuscate its goal to an observer. The main contributions of the paper are: (i) the definition of an optimization criterion that determines the cheapest set of paths that allows the actor to maintain its disclosing distance below a certain threshold; (ii) the definition of an optimization criterion that determines the set of paths that ensures that the actor achieves the best possible upper disclosing distance within a give budget; and (iii) efficient algorithms to calculate these sets.

We plan to extend this work in multiple directions. We will devise direct techniques to calculate $\lambda^{min}(C)$ in Eq. (3). We will consider *cooperative* settings in which the actor's objective is *legibility* instead of goal obfuscation and settings when only *partial information* is available.

We close by noting that path planning has often been a test bed for task planning, hence we intend to explore how the concepts investigated in this paper could be mapped into task planning.

Appendix: the proof of Theorem 5

This Appendix is devoted to the proof of the technical result Theorem 5 . We recall that, given a set of paths $\mathcal{P}$ and a value $\lambda \geq 0$, $\mathcal{H}_{\mathcal{P}}(\lambda)$ represents the graph structure on $\mathcal{P}$ that is determined by the relation of path covering at level λ indicated with $\to_\lambda$ (see Definition 3).

We start with some topological properties of the graph $\mathcal{H}_{\mathcal{P}}(\lambda)$. The first is a sort of transitivity for the relation $\leftarrow_\lambda$.

Lemma 13. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths and let $\lambda > 0$. Suppose that $\gamma_1, \dots, \gamma_n \in \mathcal{P}$ are such that*

1. $\gamma_1 \leftarrow_\lambda \gamma_2 \leftarrow_\lambda \cdots \leftarrow_\lambda \gamma_n$
2. $\gamma_k \not\leftarrow_\lambda \gamma_{k+1}$ *for any* $k = 1, \dots, n-2$
3. $\delta(\gamma_1) \neq \delta(\gamma_n)$

Then, $\gamma_1 \leftarrow_\lambda \gamma_n$

Proof Let l_k be the length of the path γ_k. Assumptions 1. and 2. guarantee the existence of points s_k for $k = 1, \dots, n-1$, such that

(a) $\gamma_k|_0^{s_k} = \gamma_{k+1}|_0^{s_k}$ for every $k \leq n-1$

(b) $W(\gamma_k|_{s_k+1}^{l_k}) \leq \lambda$ for every $k \leq n-1$

(c) $W(\gamma_{k+1}|_{s_k+1}^{l_{k+1}}) > \lambda$ for every $k \leq n-2$

From (a) and (b), it follows that $s_k < s_{k+1}$ for every $k = 1, \dots, n-1$. This implies that

$$\gamma_1|_0^{s_1} = \gamma_2|_0^{s_1} = \cdots = \gamma_n|_0^{s_1}$$

Using now Assumption 3., we deduce that $\gamma_1 \leftarrow_\lambda \gamma_n$. ■

Lemma 14. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths, $\lambda > 0$ and $\gamma_1, \dots, \gamma_n \in \mathcal{P}$ be such that*

$$\gamma_1 \leftarrow_\lambda \gamma_2 \leftarrow_\lambda \cdots \leftarrow_\lambda \gamma_n \leftarrow_\lambda \gamma_1$$

Then, at least one of the first $n-1$ relations is undirected, namely $\gamma_k \to_\lambda \gamma_{k+1}$ for some $k = 1, \dots n-1$.

Proof By contradiction, if $\gamma_k \not\leftarrow_\lambda \gamma_{k+1}$ for all $k = 1, \dots n-1$, applying Lemma 13 to the subsequence starting from γ_2, we obtain that $\lambda_2 \leftarrow_\lambda \lambda_1$. This implies the contradictory fact that $\gamma_1 \leftrightarrow_\lambda \gamma_2$. ■

Proof [of Theorem 5]

Put $\gamma_1 = \gamma$ and note first that, because of Proposition 4, there exists $\gamma_2 \in \mathcal{P}$ such that $\gamma_2 \to_{\lambda_{\gamma_1, \mathcal{P}}} \gamma_1$. Since $\lambda \geq \lambda(\mathcal{P}) \geq \lambda_{\gamma_1, \mathcal{P}}$, we also have that $\gamma_2 \to_\lambda \gamma_1$. In other terms, in the directed graph $\mathcal{H}_{\mathcal{P}}$, every node admits at least an incoming edge. In view of Lemma 14 and considering the fact that $\mathcal{P}$ is finite, starting from $\gamma_1 = \gamma$, it is always possible to find a sequence of paths $\gamma_2, \dots, \gamma_s, \gamma_{s+1}$ such that

$$\gamma_1 = \gamma \leftarrow_\lambda \gamma_2 \leftarrow_\lambda \cdots \leftarrow_\lambda \gamma_s \leftrightarrow_\lambda \gamma_{s+1}$$

where we are assuming that the step s is the first at which we meet an undirected edge, namely $\gamma_k \not\leftarrow_\lambda \gamma_{k+1}$ for any $k = 1, \dots, s-1$. If $s = 1$, then we are in the situation 1. If $s > 1$ and $\delta(\gamma_s) \neq \delta(\gamma_1)$, the assumptions of Lemma 13 are satisfied, and we deduce that $\gamma \leftarrow_\lambda \gamma_s \leftrightarrow_\lambda \gamma_{s+1}$. If instead $\delta(\gamma_s) = \delta(\gamma_1)$, necessarily it must hold that $\delta(\gamma_{s+1}) \neq \delta(\gamma_1)$ and applying again Lemma 13 on the entire sequence up to γ_{s+1} we deduce this time that $\gamma \leftarrow_\lambda \gamma_{s+1} \leftrightarrow_\lambda \gamma_s$. In both cases, we have proven condition 2. ■

References

[Chakraborti et al. 2019] Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D. E.; and Kambhampati, S. 2019. Explicability? Legibility? Predictability? Transparency? Privacy? Security? The Emerging Landscape of Interpretable Agent Behavior. In *Proc. of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS 2019, Berkeley, CA, USA*, 86–96.

[Cherkassky, Goldberg, and Radzik 1996] Cherkassky, B. V.; Goldberg, A. V.; and Radzik, T. 1996. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming* 73(2):129–174.

[Dragan, Lee, and Srinivasa 2013] Dragan, A. D.; Lee, K. C. T.; and Srinivasa, S. S. 2013. Legibility and predictability of robot motion. In *ACM/IEEE International Conference on Human-Robot Interaction, HRI 2013, Tokyo, Japan, March 3-6, 2013*, 301–308.

[Harabor and Grastien 2012] Harabor, D., and Grastien, A. 2012. The jps pathfinding system. In *Proceedings of the Fifth Annual Symposium on Combinatorial Search*, 207–208. Association for the Advancement of Artificial Intelligence (AAAI).

[Hart, Nilsson, and Raphael 1968] Hart, P.; Nilsson, N.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on* 4(2):100 –107.

[Kautz and Allen 1986] Kautz, H., and Allen, J. 1986. Generalized Plan Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 32–37.

[Keren, Gal, and Karpas 2014] Keren, S.; Gal, A.; and Karpas, E. 2014. Goal Recognition Design. In *Proceedings of the International Conference on Automated Planning and Scheduling ICAPS*.

[Keren, Gal, and Karpas 2015] Keren, S.; Gal, A.; and Karpas, E. 2015. Goal Recognition Design for Non-Optimal Agents. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 3298–3304.

[Keren, Gal, and Karpas 2016a] Keren, S.; Gal, A.; and Karpas, E. 2016a. Goal Recognition Design with Non-Observable Actions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 3152–3158.

[Keren, Gal, and Karpas 2016b] Keren, S.; Gal, A.; and Karpas, E. 2016b. Privacy preserving plans in partially observable environments. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 3170–3176.

[Kulkarni et al. 2018] Kulkarni, A.; Klenk, M.; Rane, S.; and Soroush, H. 2018. Resource Bounded Secure Goal Obfuscation. In *Proceedings of the AAAI Fall Symposium.*

[Kulkarni, Srivastava, and Kambhampati 2019] Kulkarni, A.; Srivastava, S.; and Kambhampati, S. 2019. A Unified Framework for Planning in Adversarial and Cooperative Environments. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2479–2487.

[LaValle 2006] LaValle, S. M. 2006. *Planning Algorithms.* Cambridge University Press.

[Masters and Sardiña 2017] Masters, P., and Sardiña, S. 2017. Deceptive path-planning. In *Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 4368–4375.

[Pattison and Long 2011] Pattison, D., and Long, D. 2011. Accurately determining intermediate and terminal plan states using bayesian goal recognition. In *Proceedings of the First Workshop on Goal, Activity and Plan Recognition. ICAPS*, 32–37.

[Ramirez and Geffner 2009] Ramirez, M., and Geffner, H. 2009. Plan Recognition as Planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 1778–1783.

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