An Optimization Approach to Robust Goal Obfuscation

Sara Bernardini[1] , **Fabio Fagnani**[2] , **Santiago Franco**[1]
[1]Department of Computer Science, Royal Holloway University of London
[2]Department of Mathematical Sciences, Politecnico di Torino
{sara.bernardini, santiago.franco}@rhul.ac.uk, fabio.fagnani@polito.it

Abstract

In this paper, we present a set of strategies to underpin the behavior of an agent that wants to arrive as close as possible to its destination without revealing it to an observer, which monitors its progress in the environment. This problem is an instance of goal obfuscation (GO), which has lately received significant attention in the AI community. With different variants of GO being proposed, the field lacks coherence and characterization from first principles. In addition, existing techniques are not robust to possible attempts of the observer to learn the agent's strategy. To fill this gap, we provide here a foundational study of GO and offer robust techniques to ensure that the agent can protect its privacy as much as possible regardless of the observer's behavior. We cast GO as an optimization problem, offer a complete theoretical analysis of it and introduce efficient algorithms to find exact solutions.

1 Introduction

Lately, there has been significant interest in the robotics and planning communities in interpretable agent behavior (see Chakraborti *et al.* (2019) for a survey). One of its multiple facets in adversarial settings is *goal obfuscation* (GO), which is concerned with an agent (the *actor*) that wants to maintain its goal private in the face of a second agent (the *observer*) that monitors its actions to disclose such a goal. GO techniques provides the actor with planning strategies to increase the ambiguity over the possible goals that it might want to achieve and, in consequence, help the actor protect its privacy. These techniques find applications in several fields, from surveillance to military planning to computer games, just to mention a few. Although GO takes the actor's point of view, reasoning about an agent's obfuscatory behavior can also be used to inform the strategy of an intelligent observer in applications such as search-and-tracking, where searching for an evasive target is much harder than looking for an oblivious one (Bernardini et al. 2017).

An obfuscatory actor faces a conflict between two desirable but incompatible objectives: minimizing the cost to its goal, while maximizing ambiguity over all possible goals, with the latter objective involving the use of some resources to conceal instead of to achieve the goal. Focusing on this trade-off, in this paper, we give a crisp mathematical formulation of GO and propose two, dual *optimization* problems. The first problem deals with minimizing the cost that the actor incurs to obtain a desired level of obfuscation; the second problem concerns reaching the highest possible level of obfuscation given a bound on cost. Our optimization approach allows us to perform the formulation of plans to all possible goals and the maximization of obfuscation jointly. Using the resulting set of plans, the actor is guaranteed to achieve its chosen degree of obfuscation for any goal it might pick, even if these plans become known to the observer.

We consider settings in which the actor has full observability of its environment, while the observer can either fully observe the *state* of the actor after it performs an action or not at all. This is a specific case of *partial observability* that is natural in several applicative contexts. Consider, for example, a surveillance mission in which a target aims to reach a destination that it wants to keep hidden moving on a road network and an observer tries to detect such a destination based on sensory input available only at certain locations (e.g. cameras at crossroads or search patterns centered in specific points of the network (Piacentini, Bernardini, and Beck 2019)). This type of observability represents a worst-case scenario for the actor because the observations, when successful, leave no ambiguity on the state of the actor and its advancement towards the goal. Following these assumptions, we present our results in a *path planning* setting, which represents the most natural incarnation of them, but our results hold in general for task planning when the observation function satisfies the above conditions.

Our contribution is twofold. First, we offer a foundational study of GO based on mathematical optimization and look for *exact* solutions to the proposed optimization problems that take into consideration both available resources and desired obfuscation. This is in contrast with ad hoc, approximated strategies for GO that ignore resources, which have been put forward by related works (Masters and Sardiña 2017). In addition, we propose GO techniques that are *secure* according to the definition given by Kulkarni et al. (2018). In particular, our approach is resilient to *replay attacks*: the actor is indifferent to the fact that the observer knows its algorithm and can rerun it with various goals. Existing GO techniques, on the other hand, either assume to be deployed in one-shot settings and are not robust to multiple runs of the algorithm (Keren, Gal, and Karpas 2014) or they are secure but work in settings more favorable to the actor (Kulkarni, Srivastava, and Kambhampati 2019).

2 Related Work

The majority of the work in GO has been conducted within the setting of task planning with partial observability, modeled as a many-to-one observation function that maps pairs of actions and states into observation tokens. This setting represents a more advantageous scenario for the actor than the one we consider because it forces the observer to reason in a belief state space instead of the space determined by the actual actions taken by the actor. In this context, Kulkarni et al. (2019) present a unifying framework for secure planning in both adversarial and cooperative environments. Keren et al. (2016b), on the other hand, use the *worst case distinctiveness* (wcd) index (Keren, Gal, and Karpas 2014) to indicate the maximum number of actions that the actor may take before revealing its goal. An optimal strategy for the actor is determined by selecting, for each possible goal, a legal plan to it that exhibits wcd. While in one-shot GO set-ups, this strategy is optimal in maximizing obfuscation, this is not the case in set-ups in which the observer monitors multiple, subsequent runs of the actor's algorithm. These observations could lead the observer to learn the actor's strategy. In this case, the wcd index of the selected optimal paths could be remarkably lower than the original one (see Example 4).

Reasoning about obfuscatory behavior is crucial in other 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 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. GRD, on the other hand, deals with modifying the environment in which the actor lives to facilitate or impede goal recognition. The above-mentioned work by Keren et al. (2016b) stems from GRD.

Previous work in obfuscation in path planning concerns *deceptive path planning* (Masters and Sardiña 2017). Based on a rather narrow definition of deception as ambiguity between a bogus and a real goal, they present two types of strategies, simulation and dissimulation. Simulation simply accounts for reaching a bogus destination before the real one. This strategy is expensive and not robust as it is no longer effective after the fake destination has been reached. As for dissimulation, the authors give three ad hoc, non-optimized strategies that all revolve around proceeding towards a point for which the probabilities of going to the real destination and a bogus one are equal, before switching to the real destination. No limit on agents' resources is considered. In contrast, exploring the trade-off between the desired degree of obfuscatory behavior and the resources needed to achieve it, we provide the actor with a feasible portfolio of paths that allows it to proceed as close as possible to any of the potential destinations without disclosing its intentions.

We note that there is a rich body of literature in "privacy-preserving" multi-agent planning (Torreno, Onaindia, and Sapena 2014; Brafman 2015) and path planning (Das et al. 2008; Nabil et al. 2019). However, work in these fields studies problems different from ours, making the techniques they propose not directly relevant to our case of study.

3 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 $\mathcal{H}$. The observer can see the actor only when it traverses the nodes in $\mathcal{O} \subseteq \mathcal{V}$ (*observable nodes*) and aims to discover where the actor is going. We assume that $\mathcal{H}$, o, $\mathcal{D}$, and $\mathcal{O}$ 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 in $\mathcal{H}$ 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, due to the assumptions made on the edges from the origin and into the destinations, the 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 at least 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 a set of paths $\mathcal{P}$ that is $(o, \mathcal{D})$-connecting, i.e. a set of paths that the actor can use to reach any destination in $\mathcal{D}$. We associate a *cost* to a strategy, modeled 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 either the distance along that edge or the time needed to traverse it. Conventionally, we put $W_{ij} = +\infty$ whenever $(i, j) \notin \mathcal{E}$. 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 (minimum weight of a path connecting i to j in $\mathcal{G}$). Finally, we define the cost of a path as its normalized weight, i.e. its weight relative to the geodetic distance: $C(\gamma) = \frac{W(\gamma)}{\omega(o, \delta(\gamma))}$ and $C(\mathcal{P}) = \max_{\gamma \in \mathcal{P}} C(\gamma)$. Our theory and algorithms remain unchanged if path costs are defined by using weights instead of normalized weights. Note that, when $C(\mathcal{P}) > 1$, it means that the strategy set $\mathcal{P}$ contains non-minimal paths.

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. *Two prefixes of the same length γ^k and γ'^k are said to be* **observer-equal** *if $\gamma_i = \gamma'_i$ for each $i = 0, \dots, k$ whenever γ_i or γ'_i is in $\mathcal{O}$. We use the notation $\gamma^k =_{\mathcal{O}} \gamma'^k$.*

If γ^k and γ'^k are observer-equal, they are indistinguishable from the observer's point of view.

Definition 2. *Given a set of paths $\mathcal{P}$ from o to $\mathcal{D}$, a path $\gamma \in \mathcal{P}$, and a non-negative integer t, we say that γ* **discloses** *$\mathcal{D}$ at step t if, for every $\gamma' \in \mathcal{P}$ such that $\gamma^t =_{\mathcal{O}} \gamma'^t$, it holds that $\delta(\gamma) = \delta(\gamma')$.*

Definition 2 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 3. *Given a set of paths $\mathcal{P}$ from o to $\mathcal{D}$ and a path $\gamma = (\gamma_0, \ldots, \gamma_l) \in \mathcal{P}$, 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 node at which the intentions of the actor are revealed to the observer. If γ does not disclose $\mathcal{D}$ at any step (necessarily γ leads to a destination $d \notin \mathcal{O}$), conventionally, we put $t_\gamma(\mathcal{P})$ equal to the length of γ.*
- *The* **disclosing distance** *of γ is a measure of how far the actor is from its actual destination when the observer discovers it. Formally, it is expressed as follows:*

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

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

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

The upper disclosing distance is a conservative index, which is useful when the actor wants to guarantee a certain "worst case" performance threshold: minimizing $\lambda(\mathcal{P})$ ensures that the actor's distance to any destination that it might choose in $\mathcal{D}$ is below this threshold when the chosen destination becomes clear to the observer.

Remark 4. The concept of disclosing distance introduced above is based exclusively on pairwise comparisons between paths. In certain applications, it might be interesting to consider more restrictive notions of disclosing distance, where the number of destinations compatible with a given prefix is required to be an integer q larger than 2.

Example 1. *Consider the graph $\mathcal{H}$ depicted in Fig. 1. We assume that all edges (i, j) have weight $W_{ij} = 1$. Destinations are in red and origin in green; they are observable. The other observable nodes are in black, while unobservable ones are in white. We consider two cases: when all nodes are observable $\mathcal{O} = \mathcal{V}$ (left) and when $\mathcal{O} = \{o, C, F, G, H\} \cup \mathcal{D}$ (right). We call $\mathcal{P}$ the set of all minimum distance paths in $\mathcal{H}$ from o to the four destinations $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$. Paths' disclosing indices and distances are given in Table 1. Note that $\lambda(\mathcal{P}) = 3$ when all nodes are observable, while $\lambda(\mathcal{P}) = 1$ in the partially observable case.*

$\gamma \in \mathcal{P}$	$t_\gamma(\mathcal{P})$	$\lambda_\gamma(\mathcal{P})$
oGd_1	1	1
$oEFId_2$	3	1
$oEFd_3$	3	0
$oEFCd_4$	3	1
$oABCd_4$	1	3

$\gamma \in \mathcal{P}$	$t_\gamma(\mathcal{P})$	$\lambda_\gamma(\mathcal{P})$
oGd_1	1	1
$oEFId_2$	4	0
$oEFd_3$	3	0
$oEFCd_4$	3	1
$oABCd_4$	3	1

Table 1: Disclosing indices and distances for the set of minimal distance paths $\mathcal{P}$ connecting o to $\mathcal{D}$ relative to the graph of Fig. 1 when $\mathcal{O} = \mathcal{V}$ (left) and $\mathcal{O} = \{o, C, F, G, H\} \cup \mathcal{D}$ (right).

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

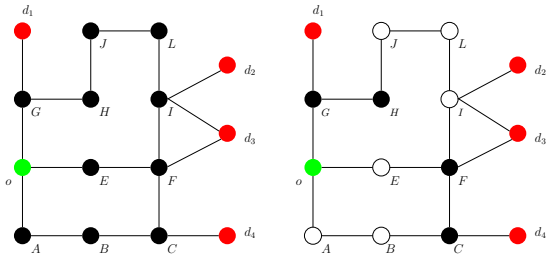

Figure 1: Example of a graph with 1 origin (in green), 4 destinations (in red) and 2 possible sets of observable nodes (in black).

1. 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,

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

2. Find a set of paths $\mathcal{P}$ 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,

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

The equations above have the following meaning. 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.

Once the strategy $\mathcal{P}$ has been decided by solving Eq. (2) or Eq. (3), the actor implements a simple strategy execution protocol: upon choosing a specific destination d, it picks one of the paths in $\mathcal{P}$ that leads to d uniformly at random. The actor can reuse its strategy multiple times to reach the same or different destinations. If the observer knows only the set of paths $\mathcal{P}$ in the strategy, it cannot hope to disambiguate the actor's destination at a distance greater than the upper disclosing one.

In Sections 4 and 5, we analyze the optimization problems (2) and (3) and propose algorithms for their solution in the special case when $\mathcal{O} = \mathcal{V}$, i.e. when all nodes are observable. From the agent's point of view, this is a worst case scenario. In Section 6, however, we show that this assumption does not entail any loss of generality, as we can always transform a graph $\mathcal{H}$ with any number of unobservable nodes to a new graph $\mathcal{H}^{\mathcal{O}}$ where all nodes are observable and that is equivalent to $\mathcal{H}$ for the purpose of solving problems (2) and (3).

4 Path Covering Properties

We now undertake an in-depth study of the notion of upper disclosing distance introduced in Definition 3. 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 recall our standing assumption that $\mathcal{O} = \mathcal{V}$. We start with a pairwise relation that is at the core of the concept of disclosing distance.

Definition 5. *Given two paths γ_1 and γ_2 from o to $\mathcal{D}$, we say that γ_2* **covers** *γ_1 at level $\lambda \geq 0$ (and write $\gamma_2 \to_\lambda \gamma_1$) iff:*

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

When both $\gamma_2 \to_\lambda \gamma_1$ and $\gamma_1 \to_\lambda \gamma_2$, we write $\gamma_2 \leftrightarrow_\lambda \gamma_1$.

Definition 5 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. The following result clarifies how this covering concept is connected with the disclosing distance.

Proposition 6. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths and let $\gamma \in \mathcal{P}$. Then, for every $\lambda \geq 0$,*

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

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 $\lambda_\gamma(\mathcal{P}) = W(\gamma|_{t_\gamma(\mathcal{P})}^l) \leq W(\gamma|_t^l) \leq \lambda$.

$\Leftarrow$: Proven similarly, inverting the steps above. ■

The following result plays a crucial role in 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 relationships involving either pairs or triples of paths exist among them.

Theorem 7. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths and let $\lambda(\mathcal{P}) \leq \lambda$. 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 $\gamma'' \leftrightarrow_\lambda \gamma' \to_\lambda \gamma$.*

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

We now introduce some notation that is helpful to prove the results that follow and to illustrate our algorithms. 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 ∂N_d^λ;
- $\overline{N}_{d_1,d_2}^\lambda = \overline{N}_{d_1}^\lambda \cap \overline{N}_{d_2}^\lambda, \quad N^{2,\lambda} = \bigcup_{(d_1,d_2) \in \mathcal{D} \times \mathcal{D}} \overline{N}_{d_1,d_2}^\lambda$;
- $\widetilde{\partial N}_d^\lambda = \partial N_d^\lambda \setminus N^{2,\lambda}$ 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 $N^{2,\lambda}$.

Definition 8. *Two distinct destinations $d_1 \neq d_2$ are called* λ-**interacting** *if $\overline{N}_{d_1,d_2}^\lambda \neq \varnothing$.*

We further define:

$$\mathcal{D}_\lambda^2 := \{(d_1, d_2) \in \mathcal{D} \times \mathcal{D} \,|\, d_1,\ d_2\ \lambda\text{-interacting}\}$$
$$\mathcal{D}_\lambda^1 := \{d \in \mathcal{D} \,|\, \exists d' \in \mathcal{D} \,|\, (d,\ d') \in \mathcal{D}_\lambda^2\}$$
$$\lambda^* = \min\{\lambda \geq 0 \,|\, \mathcal{D}_\lambda^2 \neq \varnothing\}$$

We now illustrate these objects in a simple example.

Example 2. *In Figure 2, we consider a 10×10 grid (4-degree 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 closed neighbors $\overline{N}_{d_i}^1$ for $i = 1, \ldots, 4$, while the area shown in a darker color represents the intersection $\overline{N}_{d_2,d_3}^1$. The parts of the closed neighborhoods of d_1 and d_4 within the dashed perimeters are the 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. Ignore the paths shown in the figures for now. Note that d_2 and d_3 are 1-interacting and no other pairs in the graph is 1-interacting. Hence, in this case, $\mathcal{D}_1^2 = \{(d_2, d_3), (d_3, d_2)\}$ and $\mathcal{D}_1^1 = \{d_2, d_3\}$. 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 7, we obtain the following fundamental limitation on the disclosing distance that the actor can achieve, regardless of the budget at its disposal.

Corollary 9. *Let $\mathcal{P}$ be an $(o, \mathcal{D})$-connecting set of paths. Then, $\lambda(\mathcal{P}) \geq \lambda^*$*

Proof Fix $\lambda = \lambda(\mathcal{P})$ and notice that Theorem 7 implies the existence of a pair of paths γ' and γ'' in $\mathcal{P}$ of length, respectively, l' and l'', such that $\gamma' \leftrightarrow_\lambda \gamma''$. 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 $\gamma'_h = \gamma''_h \in \overline{N}_{\delta(\gamma')}^\lambda \cap \overline{N}_{\delta(\gamma'')}^\lambda$. Therefore, $\mathcal{D}_\lambda^2 \neq \varnothing$ and, given the definition of λ^*, it follows that $\lambda \geq \lambda^*$. ■

The construction proposed in the next section shows that the inequality in Corollary 9 is sharp: the limit λ^* can always be achieved if a sufficient budget is available.

5 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 $\lambda \geq \lambda^*$.

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

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

Finally, we define:

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

Example 3. *In Figure 2, we show two examples of our canonical construction for the graph of Example 2. On the left, we represent the optimal pair of paths $\mathcal{P}_{d_2,d_3}^1$ with the bifurcation node $h^* \in \overline{N}_{d_2,d_3}^1$. On the right, we represent the optimal triple of paths $\mathcal{P}_{d_4}^1 = \mathcal{P}_{d_4,d_2,d_3}^{(h_1^*,h_2^*)}$ with the two bifurcation nodes $h_1^* \in \partial N_{d_4}^1 = \widetilde{\partial N}_{d_4}^1$ and $h_2^* \in \overline{N}_{d_2,d_3}^1$.*

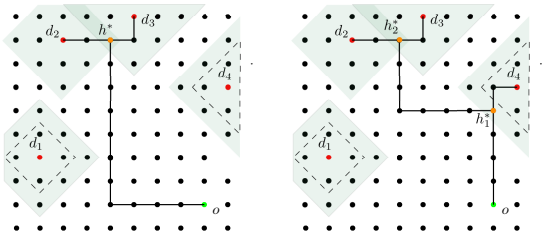

Figure 2: Two examples of the canonical construction.

The protection pattern involving pairs of paths (see Figure 2-left) can be thought as an implementation of a *dissimulation* strategy. We can protect pairs of destinations that are close to each other by choosing a path that initially is ambiguously compatible with both and bifurcates only when it gets sufficiently close to them. In this way, the two destinations reciprocally protect themselves. On the other hand, the protection pattern involving triples of paths (see Figure 2-right) can be seen as an implementation of a *simulation* strategy. We can protect a destination that is isolated from the others, by selecting a path that first goes unambiguously in its direction and then, once close by, bifurcates with one branch going to it and another branch moving to a pair of other destinations. From the bifurcation, the path becomes ambiguous. In this case, the pair of destinations protects the isolated one. If pairs of sufficiently close destinations do not exist, the required level of obfuscation is impossible to obtain (see Corollary 9).

The next example shows the power of our technique with respect to other approaches. Our construction is resistant to replay attacks (potential observer's learning attempts).

Example 4. *Figure 3-left shows the complete set of paths $\mathcal{P}^1$ for the graph of Example 2. Consider now the set of paths $\tilde{P}$ that, starting from all possible paths γ from o to $\mathcal{D}$ with $C(\gamma) \le C(\mathcal{P}^1)$, individually maximizes the* wcd-$d_i(\mathcal{D})$ *for $i = 1, \ldots, 4$ as defined by Keren* et al. *(2016b) (Figure 3-right). If the observer learns $\tilde{P}$ via multiple observations, obfuscation becomes as low as* wcd-$d_4(\mathcal{D}) = 0$. *Instead, by using the equally costed path set $\mathcal{P}^1$,* wcd-$d_4(\mathcal{D})$ *always remains 4 regardless of the observer's learning efforts.*

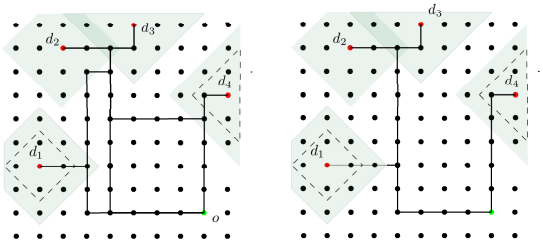

Figure 3: The set of paths $\mathcal{P}^1$ (left) and $\tilde{P}$ (right).

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

Theorem 10. *Let $\lambda \ge \lambda^*$. Then, $\mathcal{P}^\lambda$ is $(o, \mathcal{D})$-connecting and $\lambda(\mathcal{P}^\lambda) \le \lambda$.*

Proof Note first that, based on the definition of λ^*, we have that $\mathcal{D}_\lambda^2$ is not empty. From this fact and the definition of $\mathcal{P}_d^\lambda$, it follows that, for any destination $d \in \mathcal{D}$, it holds that $\mathcal{P}_d^\lambda \neq \emptyset$. This proves that $\mathcal{P}^\lambda$ is $(o, \mathcal{D})$-connecting. Finally, the fact that $\lambda(\mathcal{P}^\lambda) \le \lambda$ follows from Proposition 6. ■

We now present the main result of this section, which shows the universal role of the set of paths $\mathcal{P}^\lambda$. The theorem asserts that, given a set of paths whose upper disclosing distance is below a certain threshold λ, the same performance with the same budget can be obtained by a suitable subset of $\mathcal{P}^\lambda$. Therefore, when searching for a set of paths of minimal cost that attains a given threshold λ, we can carry out the search within this $\mathcal{P}^\lambda$.

Theorem 11. *Given $\lambda \ge 0$, for every set of paths $\mathcal{P}$ that is $(o, \mathcal{D})$-connecting and such that $\lambda(\mathcal{P}) \le \lambda$, there exists $\mathcal{Q} \subseteq \mathcal{P}^\lambda$ that is $(o, \mathcal{D})$-connecting and such that:*

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

Proof We construct $\mathcal{Q}$ as follows. 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 $\delta(\gamma) = \delta(\tilde{\gamma})$. $\mathcal{Q}_\gamma$ satisfies the following two properties:

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

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., $C(\mathcal{Q}) \le$

$C(\mathcal{P})$. Because of property 1., for each $\sigma \in \mathcal{Q}_\gamma$, there exists $\sigma' \in \mathcal{Q}_\gamma$ such that $\sigma' \rightarrow_\lambda \sigma$. 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 via Lemma 7.

Suppose $\gamma \in \mathcal{P}$ with $d = \delta(\gamma)$ satisfies condition 1. of Theorem 7, i.e. there exists $\gamma' \in \mathcal{P}$ such that $\gamma' \leftrightarrow_\lambda \gamma$. In this case, we pick $\mathcal{Q}_\gamma = \mathcal{P}^\lambda_{d,d'}$ where $d' = \delta(\gamma')$. Property 1. is automatically verified. To prove property 2. 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 $\mathcal{P}^\lambda_{d,d'}$ 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 Theorem 7 does not hold for γ. Then, condition 2. must hold. Hence, there exist $\gamma' \in \mathcal{P}$ and $\gamma'' \in \mathcal{P}$ such that $\gamma'' -_\lambda \gamma' \rightarrow_\lambda \gamma$. We put $d' = \delta(\gamma')$ and $d'' = \delta(\gamma'')$. 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 7 would instead hold true. If $d \notin \{d', d''\}$, by the definition of $\mathcal{P}^\lambda_d$ 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

$$C(\mathcal{P}^\lambda_d) \leq C(\{\gamma^{o,h,d}, \gamma^{o,\tilde{h},h',d'}, \gamma^{o,\tilde{h},h',d''}\}) \leq C(\gamma, \gamma', \gamma'')\} \leq C(\mathcal{P})$$

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 $C(\gamma^{o,h',d''}) \leq C(\gamma^{o,\tilde{h},h',d''})$. The proof is now complete. ■

Theorem 11 implies that, when searching for a minimum of the problem in Eq. (2), we can restrict to subsets of $\mathcal{P}^\lambda$.

Corollary 12. *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 11, 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). ■

Remark 13. We measure the obfuscation level by means of the upper disclosing distance $\lambda(\mathcal{P})$, which is calculated considering the entire set of destinations. This is a global, "worst case" index that does not take into consideration the level of obfuscation achieved for the single destinations. However, it is worth noting that our canonical construction is also optimal for destination specific indexes, defined as:

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

Indeed, it follows from the proof of Theorem 11 that the set $\mathcal{Q}$ also satisfies $\lambda(\mathcal{Q}, d) \leq \lambda(\mathcal{P}, d)$ for every possible destination $d \in \mathcal{D}$.

5.2 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 as follows. 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:

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

If the set over which we calculate the minimum is empty, we conventionally 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 problem in Eq. (2):

Corollary 14. $C(\mathcal{P}^{\lambda\,\text{opt}}) = C^{min}(\lambda)$

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 7, two paths $\gamma', \gamma'' \in \mathcal{Q}$ such that $\gamma'' \leftrightarrow_\lambda \gamma' \rightarrow_\lambda \gamma$. 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 12, the proof is complete. ■

We now make some comments on this construction.

1. Notice first that, when there is only one λ-interacting pair, $\mathcal{P}^{\lambda\,\text{opt}}$ coincides with $\mathcal{P}^\lambda$. For the 10×10 grid analyzed in Example 2, the set of paths represented in Figure 3 is therefore, in that case, the optimal $\mathcal{P}^{1\,\text{opt}}$.
2. In the general case, the cardinality of $\mathcal{P}^{\lambda\,\text{opt}}$ satisfies the bound $|\mathcal{P}^{\lambda\,\text{opt}}| \leq 3|\mathcal{D}|$.

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 4. 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 $\mathcal{P}^2_{d_4}$.

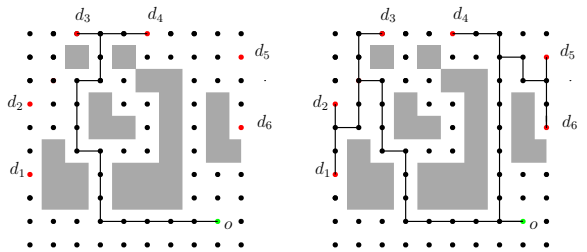

Figure 4: Left: Non-optimal choice for d_3 and for d_4. Right: Optimal choices $\mathcal{P}^2_{d_3}$ and $\mathcal{P}^2_{d_4}$.

In a practical implementation of this algorithm (Algorithm 1), the construction of the optimal set of paths $\mathcal{P}^{\lambda\,\mathrm{opt}}$ can be done directly, without constructing $\mathcal{P}^\lambda$ first.

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** $\omega(o,d) \le \lambda$ **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** $\omega(o,d) \le \lambda$ **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** $\omega(o,d) \le \lambda$ **then**
19 **continue**;
20 **foreach** $(d', d'') \in \mathcal{D}^2_\lambda \; s.t. \; d \neq d', d''$ **do**
21 **foreach** $h_1 \in \overline{\partial N}^\lambda_d$ **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** $\mathcal{P}^{\lambda\mathrm{opt}}$;

The algorithm works in three steps:

1. For destinations d such that $\omega(o,d) \le \lambda$, the algorithm 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 that (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 nodes, between the different bifurcation nodes, and between the bifurcation nodes 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 case scenario where, to construct $\mathcal{P}^{\lambda\,\mathrm{opt}}_d$, all possible pairs $\mathcal{P}^{(h)}_{d,d'}$ (with $d' \neq d$) and 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 topological 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:

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

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 $\lambda^{min}(C)$ that 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.

6 Partial observability

This section is devoted to the partial observability case, i.e. when $\mathcal{O} \neq \mathcal{V}$. We show that we can always transform the original graph with unobservable nodes to a new graph with observable nodes only and that the two graphs are equivalent with respect to solving the optimization problems (2) and (3). Hence, we can apply the theory and algorithms developed above for the fully observable case to the partially observable case. Below, we describe this important transformation step.

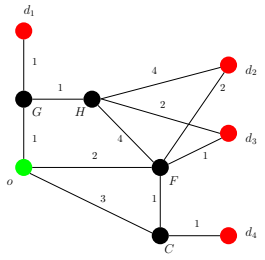

Figure 5: $\mathcal{H}^{\mathcal{O}}$

$\sigma \in \mathcal{P}^{\mathcal{O}}$	$t_\sigma(\mathcal{P}^{\mathcal{O}})$	$\lambda_\sigma(\mathcal{P}^{\mathcal{O}})$
oGd_1	1	1
oFd_2	2	0
oFd_3	2	0
$oFCd_4$	2	1
oCd_4	1	1

Table 2: The set $\mathcal{P}^{\mathcal{O}}$ with disclosing indices and distances.

We first note that, because of the way the disclosing distances have been defined, it does not entail any loss of generality to assume that $\{o\} \cup \mathcal{D} \subseteq \mathcal{O}$. We now define a new graph $\mathcal{H}^{\mathcal{O}} = (\mathcal{O}, \mathcal{F})$ where the edge set $\mathcal{F}$ is defined as follows. Given a pair of nodes $v, w \in \mathcal{O}$, consider the set of paths $\Gamma_{v,w}$ that connect v to w in $\mathcal{H}$ and do not pass through observable nodes at the intermediate steps. Then, $\mathcal{F} = \{(v,w) \in \mathcal{O} \times \mathcal{O} \,|\, \Gamma_{v,w} \neq \varnothing\}$. We also define a weight $W^{\mathcal{O}}$ on this new graph, by putting, for every $(v,w) \in \mathcal{F}$, $W^{\mathcal{O}}_{vw} = \min\{W(\gamma) \,|\, \gamma \in \Gamma_{v,w}\}$. For future use, for every $(v,w) \in \mathcal{F}$, we fix a priori a path $\gamma_{v,w} \in \Gamma_{v,w}$ of minimal length, i.e. such that $W(\gamma_{v,w}) = W^{\mathcal{O}}_{vw}$.

We can associate a path $\gamma^{\mathcal{O}}$ on the new graph $\mathcal{H}^{\mathcal{O}}$ to every path γ connecting the origin o to a destination d in the original graph $\mathcal{H}$ by simply deleting all nodes that are not in $\mathcal{O}$. $\gamma^{\mathcal{O}}$ still connects o to d in $\mathcal{H}^{\mathcal{O}}$. Conversely, given any path $\sigma = (\sigma_0, \ldots, \sigma_m)$ connecting the origin o to a destination d in the new graph $\mathcal{H}^{\mathcal{O}}$, we associate to σ a path $\sigma^{\mathcal{V}}$ that connects o to d in the old graph $\mathcal{H}$ defined as follows: $\sigma^{\mathcal{V}} = \gamma_{\sigma_0,\sigma_1} \perp \gamma_{\sigma_1,\sigma_2} \perp \ldots \perp \gamma_{\sigma_m - 1, \sigma_m}$. Note that $(\sigma^{\mathcal{V}})^{\mathcal{O}} = \sigma$ but in general $(\gamma^{\mathcal{O}})^{\mathcal{V}} \neq \gamma$. Given the sets of paths $\mathcal{P}$ and $\mathcal{Q}$ from o to $\mathcal{D}$ in, respectively, $\mathcal{H}$ and $\mathcal{H}^{\mathcal{O}}$, we put $\mathcal{P}^{\mathcal{O}} = \{\gamma^{\mathcal{O}} \,|\, \gamma \in \mathcal{P}\}$ and $\mathcal{Q}^{\mathcal{V}} = \{\sigma^{\mathcal{V}} \,|\, \gamma \in \mathcal{Q}\}$.

Example 5. *Consider the graph $\mathcal{H}$ of Example 1 with $\mathcal{O} = \{o, C, F, G, H\} \cup \mathcal{D}$, depicted on the right in Fig. 1. $\mathcal{P}$ is the set of all minimum distance paths in $\mathcal{H}$ from o to $\mathcal{D}$ whose elements were listed on the right in Table 1. In Fig. 5, we depict the graph $\mathcal{H}^{\mathcal{O}}$ and, in Table 2, we give the list of elements in $\mathcal{P}^{\mathcal{O}}$ together with their disclosing indices and distances. Note that $\lambda(\mathcal{P}^{\mathcal{O}}) = 1$.*

The following result asserts that our optimization problems (2) and (3) on $\mathcal{H}$ with partial observability can equivalently be solved on the transformed $\mathcal{H}^{\mathcal{O}}$ where observability is full. Its proof is postponed to Appendix B.

Theorem 15. *The optimization problems (2) and (3) attain the same minimum on the two graphs $\mathcal{H}$ and $\mathcal{H}^{\mathcal{O}}$. Also,*

$$\begin{aligned} &\mathcal{P} \text{ minimum for } \mathcal{H} \;\Rightarrow\; \mathcal{P}^{\mathcal{O}} \text{ minimum for } \mathcal{H}^{\mathcal{O}} \\ &\mathcal{Q} \text{ minimum for } \mathcal{H}^{\mathcal{O}} \;\Rightarrow\; \mathcal{Q}^{\mathcal{V}} \text{ minimum for } \mathcal{H} \end{aligned} \tag{8}$$

This result states that, from the theoretical point of view, it does not entail any loss of generality to assume that $\mathcal{O} = \mathcal{V}$, i.e. that all nodes are observable.

6.1 Algorithm to Compute $\mathcal{H}^{\mathcal{O}}$

In what follows, we propose an algorithmic construction of the graph $\mathcal{H}^{\mathcal{O}} = (\mathcal{O}, \mathcal{F})$ and the weight matrix $W^{\mathcal{O}}$.

The algorithm works in three steps:

1. It puts in $\mathcal{F}$ all edges (v,w) in the original graph $\mathcal{H}$ that connect pairs of nodes in $\mathcal{O}$ and temporarily give them the same weight as they had in $\mathcal{H}$ (lines 4 to 8);
2. For each node $v \in \mathcal{O}$, it constructs a subgraph of $\mathcal{H}$ by trimming all outgoing edges from nodes in $\mathcal{O}$ except those edges from v to nodes in $\mathcal{V} \setminus \mathcal{O}$. Then, by using Dijkstra's algorithm, the algorithm finds the minimum cost paths from v to all other reachable nodes (lines 10 to 16);
3. When there is a path from v to some node $w \in \mathcal{O}$, it puts (v,w) in $\mathcal{F}$ and updates the weight matrix W if the path's cost is inferior to the cost of the direct edge (v,w), if it exists (lines 17 to 20).

Algorithm 2: Calculation of $\mathcal{H}^{\mathcal{O}}$

```
   Input: H = (O, E), W
   Output: H^O = (O, F), W^O
1  TempEdges = ∅;
2  TempOrigins = ∅;
3  ObsToUnobsEdges = ∅;
4  foreach (v, w) ∈ E do
5      if v ∈ O then
6          if w ∈ O then
7              F ← (v, w);
8              W^O_vw = W_vw;
9          else
10             ObsToUnobsEdges ← (v, w);
11             TempOrigins ← v
12     else
13         TempEdges ← (v, w)
14 foreach v ∈ TempOrigins do
15     foreach (v, w) ∈ ObsToUnobsEdges do
16         TempEdges ← (v, w);
17     Γ = Dijkstra(v, TempEdges);
18     foreach γ = (γ_0, ..., γ_l) ∈ Γ do
19         if γ_l ∈ O then
20             if W(γ) < W_vγ_l then
21                 F ← (v, γ_l);
22                 W^O_vγ_l = W(γ);
23     foreach (v, w) ∈ ObsToUnobsEdges do
24         TempEdges \ (v, w);
```

7 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 boundaries nodes). 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 (e.g. Shanghai) and are represented as 256×256 square grid graphs with obstacles and each node having degree 8 (except for boundaries nodes). 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.

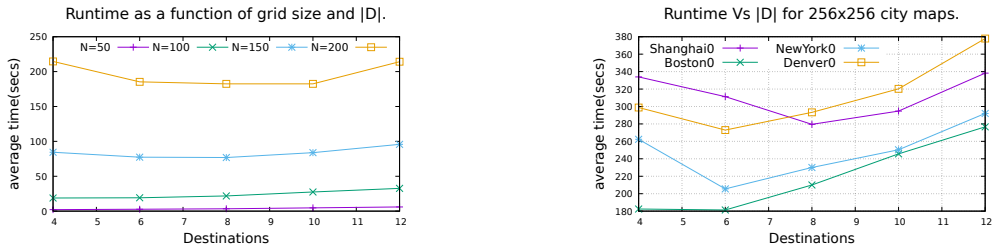

Figure 6: Runtime of Algorithm 1 for grid graphs (left) and city fragments from 'Moving-AI' benchmarks.

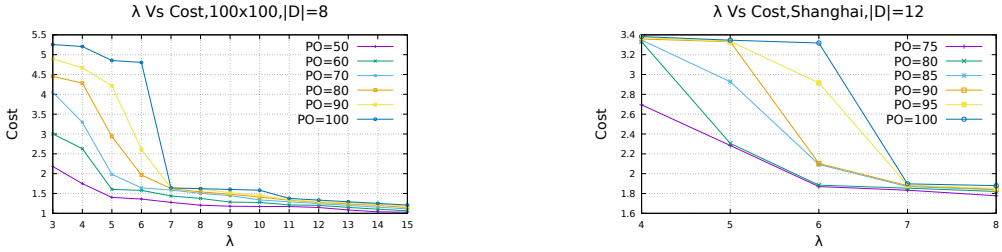

Figure 7: Cost $C^{min}(\lambda)$ for various $\lambda \geq \lambda^*$ and PO rates.

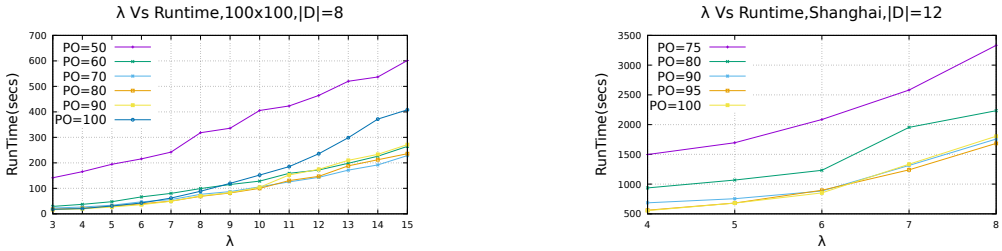

Figure 8: Algorithm's runtime for different λ and PO rates.

For both groups of problems, we performed three series of experiments. The first series (Figure 6) focuses on how the runtime (in secs) of Algorithm 1 changes as a function of the graph size (for grids) or configuration (for cities) and the number of destinations in the full observability case. 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.

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. We indicate with *PO* the percentage of observable nodes. We randomly select the number of observable nodes, and each data point is the average of 10 randomized selections with different seeds. We see that a dramatic drop in the cost happens when λ approaches a certain threshold value. This happens when new bifurcations, cheaper to reach, can be used to cover the different destinations. The figure also shows that the more unobservable nodes, the cheapest is for the actor to obfuscate. This is unsurprising: unobservable nodes allow the actor to reach destinations via paths whose λ would be higher if all nodes were observable.

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. With total observability, it runs in around 400 seconds for the biggest λ for the 100x100 grid and around 30 minutes for Shanghai. In most of the cases, the lower the PO rate, the higher the runtime. This is due to the fact that, despite the reduction in size of the observable graph $\mathcal{H}^{\mathcal{O}}$, its construction becomes increasingly more expensive as the Dijkstra search need to be performed on graphs with an increased number

of edges (Algorithm 2, line 17).

8 Conclusions and Future Work

In this paper, we rigorously define GO in terms of two optimization problems that allow the actor to balance its available resources with its desired level of obfuscatory behavior. We consider a specific case of partial observability: when the actor takes an action, the observer either fully observes the resulting state or not at all. This case is particularly unfavorable to the actor as the observer does not need to reason within a belief search space and it is a natural assumption in path-planning domains. In contrast with existing methods, we present algorithms to tackle this case and offer the actor solutions that are exact and resilient to replay attacks by the observer. Building on this foundational work, we will tackle domains with different observation functions in future work.

Appendix A: Proof of Theorem 7

We first note that the covering relation $\gamma_1 \rightarrow_\lambda \gamma_2$ (Definition 5) naturally leads to a directed graph structure on the set of paths from o to $\mathcal{D}$, which we denote with the symbol $\mathcal{H}_\mathcal{P}(\lambda)$. The graph $\mathcal{H}_P(\lambda)$ is monotonically increasing in λ.

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 16. *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 (b) and (c), it follows that $s_k < s_{k+1}$ for every $k = 1, \dots, n-2$. 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 17. *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 $\gamma_k \rightarrow_\lambda \gamma_{k+1}$ 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 16 to the subsequence starting from γ_2, we obtain that $\gamma_2 \leftarrow_\lambda \gamma_1$. This implies the contradictory fact that $\gamma_1 \leftrightarrow_\lambda \gamma_2$. ■

Proof [of Theorem 7] Put $\gamma_1 = \gamma$ and note first that, because of Proposition 6, there exists $\gamma_2 \in \mathcal{P}$ such that $\gamma_2 \rightarrow_{\lambda_{\gamma_1,\mathcal{P}}} \gamma_1$. Since $\lambda \geq \lambda(\mathcal{P}) \geq \lambda_{\gamma_1,\mathcal{P}}$, we also have that $\gamma_2 \rightarrow_\lambda \gamma_1$. In other terms, in the directed graph $\mathcal{H}_\mathcal{P}$, every node admits at least an incoming edge. In view of Lemma 17 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 16 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 16 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. ■

Appendix B: Proof of Theorem 15

Proposition 18. *Given the set of paths $\mathcal{P}$ and $\mathcal{Q}$ from o to $\mathcal{D}$ in, respectively, $\mathcal{H}$ and $\mathcal{H}^\mathcal{O}$, the following relations hold:*

1. $C(\mathcal{P}^\mathcal{O}) \leq C(\mathcal{P})$, $C(\mathcal{Q}^\mathcal{V}) = C(\mathcal{Q})$
2. $\lambda(\mathcal{P}^\mathcal{O}) \leq \lambda(\mathcal{P})$, $\lambda(\mathcal{Q}^\mathcal{V}) = \lambda(\mathcal{Q})$

Proof By construction, for every $\gamma \in \mathcal{P}$ and $\sigma \in \mathcal{Q}$, it holds $W^\mathcal{O}(\gamma^\mathcal{O}) \leq W(\gamma)$ and $W^\mathcal{O}(\sigma) = W(\sigma^\mathcal{V})$. This yields 1.

We now prove the first inequality in item 2. Consider a path $\gamma \in \mathcal{P}$ of length l and put $t = t_\gamma(\mathcal{P})$. There exists $\gamma' \in \mathcal{P}$ such that $\delta(\gamma) \neq \delta(\gamma')$, $\gamma^{t-1} =_\mathcal{O} \gamma'^{t-1}$ and, necessarily, $\gamma_t \in \mathcal{O}$. Consider $\gamma^\mathcal{O} = (\gamma_0, \gamma_{k_1}, \dots, \gamma_{k_m})$ and let $s \in \{1, \dots, m\}$ be such that $k_s = t$. Then, $(\gamma^\mathcal{O})^{s-1} = (\gamma'^\mathcal{O})^{s-1}$ and thus $t_{\gamma^\mathcal{O}}(\mathcal{P}^\mathcal{O}) \geq s$. This implies $\lambda_{\gamma^\mathcal{O}}(\mathcal{P}^\mathcal{O}) \leq W^\mathcal{O}(\gamma_s^\mathcal{O}, \dots, \gamma_m^\mathcal{O}) \leq W(\gamma_t, \dots, \gamma_l) = \lambda_\gamma(\mathcal{P})$ This yields $\lambda(\mathcal{P}^\mathcal{O}) \leq \lambda(\mathcal{P})$. We now prove the second relation in item 2. Consider a path $\sigma \in \mathcal{Q}$ of length l and put $t = t_\sigma(\mathcal{P}^\mathcal{O})$. There exists $\sigma' \in \mathcal{Q}$ such that $\delta(\sigma) \neq \delta(\sigma')$, $\sigma^{t-1} = \sigma'^{t-1}$ and $\sigma_t \neq \sigma'_t$. Let s be the length of the prefix $\gamma_{\sigma_0,\sigma_1} \perp \gamma_{\sigma_1,\sigma_2} \perp \dots \perp \gamma_{\sigma_{t-1},\sigma_t}$. Since $(\sigma^\mathcal{V})^{s-1} =_\mathcal{O} (\sigma'^\mathcal{V})^{s-1}$, it follows that $t_{\sigma^\mathcal{V}}(\mathcal{Q}^\mathcal{V}) \geq s$ and, thus, $\lambda_{\sigma^\mathcal{V}}(\mathcal{Q}^\mathcal{V}) \leq W(\gamma_{\sigma_t,\sigma_t+1} \perp \dots \perp \gamma_{\sigma_{l-1},\sigma_l}) = W^\mathcal{O}(\sigma_t, \dots, \sigma_l) = \lambda_\sigma(\mathcal{Q})$. This yields $\lambda(\mathcal{Q}^\mathcal{V}) \leq \lambda(\mathcal{Q})$. To show that the equality actually holds, note that from the first relation proven and the fact that $\mathcal{Q} = (\mathcal{Q})^\mathcal{V})^\mathcal{O}$, we have that $\lambda(\mathcal{Q}) = \lambda((\mathcal{Q})^\mathcal{V})^\mathcal{O}) \leq \lambda(\mathcal{Q}^\mathcal{V})$. The proof is now complete. ■

Proof [of Theorem 15] Consider the problem (2) and put $C^{\min}(\lambda)$ and $C^{\mathcal{O}\min}(\lambda)$ the minimum attained for the two graphs $\mathcal{H}$ and $\mathcal{H}^\mathcal{O}$, respectively. For a fixed λ, suppose $\mathcal{P}$ is an $(o, \mathcal{D})$-connecting set of paths in $\mathcal{H}$ such that $\lambda(\mathcal{P}) \leq \lambda$ and it reaches the minimum of (2), i.e. $C^{\min}(\lambda) = C(\mathcal{P})$. Then, $\mathcal{P}^\mathcal{O}$ is an $(o, \mathcal{D})$-connecting set of paths in $\mathcal{H}^\mathcal{O}$ and $\lambda(\mathcal{P}^\mathcal{O}) \leq \lambda$ due to item 2. of Proposition 18. Also, due to item 1. of Proposition 18, $C(\mathcal{P}^\mathcal{O}) \leq C(\mathcal{P}) = C^{min}(\lambda)$. This implies that $C^{\mathcal{O}\min}(\lambda) \leq C^{\min}(\lambda)$. Arguing on the dual transformation $\mathcal{Q} \rightarrow \mathcal{Q}^\mathcal{V}$, we instead obtain that $C^{\min}(\lambda) \leq C^{\mathcal{O}\min}(\lambda)$. This proves equality and the validity of implications (8). A similar proof works for problem (3). ■

Acknowledgements

The authors thank Sarah Keren for her helpful comments. This work has been supported by EPSRC Grant

EP/S016473/1 and MIUR Grant "Dipartimenti di Eccellenza 2018-2022" (CUP: E11G18000350001).

References

Bernardini, S.; Fox, M.; Long, D.; and Piacentini, C. 2017. Deterministic vs probabilistic methods for searching for an evasive target. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*.

Brafman, R. I. 2015. A privacy preserving algorithm for multi-agent planning and search. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 1530–1536. AAAI Press.

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.

Das, A. S.; Keshri, J. K.; Srinathan, K.; and Srivastava, V. 2008. Privacy preserving shortest path computation in presence of convex polygonal obstacles. In *2008 Third International Conference on Availability, Reliability and Security*, 446–451.

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

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

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

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.

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.

Nabil, M.; Sherif, A.; Mahmoud, M.; Alsharif, A.; and Abdallah, M. 2019. Efficient and privacy-preserving ridesharing organization for transferable and non-transferable services. In *EEE Transactions On Dependable and Secure Computing*, 1–16.

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.

Piacentini, C.; Bernardini, S.; and Beck, C. 2019. Autonomous Target Search with Multiple Coordinated UAVs. *Journal of Artificial Intelligence Research* 65:519–568.

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

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

Torreno, A.; Onaindia, E.; and Sapena, O. 2014. Distributed cooperative multi-agent planning. *Applied Intelligence* 41(2):606–626.