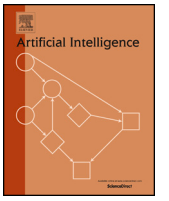
Artificial Intelligence 333 (2024) 104148

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Optimizing pathfinding for goal legibility and recognition in cooperative partially observable environments

Sara Bernardini [a,b,*], Fabio Fagnani [c], Alexandra Neacsu [a], Santiago Franco [a]

[a] Royal Holloway University of London, Egham Hill, TW20 0EX, Egham, United Kingdom
[b] University of Rome "La Sapienza", Piazzale Aldo Moro, 5, Roma, 00185, Italy
[c] Politecnico di Torino, Corso Duca degli Abruzzi, Torino, 10129, Italy

ARTICLE INFO

Keywords:
Goal legibility
Goal recognition
Interpretable agent behavior
Multi-agent path planning
Pathfinding
Partial observability
Flow networks
Reformulation

ABSTRACT

In this paper, we perform a joint design of goal legibility and recognition in a cooperative, multi-agent pathfinding setting with partial observability. More specifically, we consider a set of identical agents (the actors) that move in an environment only partially observable to an observer in the loop. The actors are tasked with reaching a set of locations that need to be serviced in a timely fashion. The observer monitors the actors' behavior from a distance and needs to identify each actor's destination based on the actor's observable movements. Our approach generates legible paths for the actors; namely, it constructs one path from the origin to each destination so that these paths overlap as little as possible while satisfying budget constraints. It also equips the observer with a goal-recognition mapping between unique sequences of observations and destinations, ensuring that the observer can infer an actor's destination by making the minimum number of observations (legibility delay). Our method substantially extends previous work, which is limited to an observer with full observability, showing that optimizing pathfinding for goal legibility and recognition can be performed via a reformulation into a classical minimum cost flow problem in the partially observable case when the algorithms for the fully observable case are appropriately modified. Our empirical evaluation shows that our techniques are as effective in partially observable settings as in fully observable ones.

1. Introduction

In the AI and robotics literature, the concept of *goal legibility* emerges in settings where there are two types of agents at play: the artificial agents that act in the environment to achieve specific goals, called the *actors*, and the agents (either human or artificial) that monitor the actors' actions to identify their goals, called the *observers*. Often, there are multiple actors and one observer. The observer runs a *goal recognition* system, i.e. an algorithm that aims to recognize the objective that an actor is trying to achieve based on a sequence of observations of the actor's actions. An actor is said to display a *legible* behavior when it is able to signal the goal it is pursuing to the observer through its actions, with *no direct communication*, which might not always be available. The notion of legibility arises in *cooperative* or *collaborative* domains where the actor is aware of the observer's presence and wants to make it easy for it to understand its objective.

* Corresponding author at: Royal Holloway University of London, Egham Hill, TW20 0EX, Egham, United Kingdom.
E-mail addresses: Sara.Bernardini@rhul.ac.uk (S. Bernardini), fabio.fagnani@polito.it (F. Fagnani), Alexandra.Neacsu.2019@live.rhul.ac.uk (A. Neacsu), Santiago.FrancoAixela@rhul.ac.uk (S. Franco).

https://doi.org/10.1016/j.artint.2024.104148
Received 13 April 2023; Received in revised form 3 May 2024; Accepted 11 May 2024

Available online 21 May 2024

Legibility has initially emerged in human-robot interaction settings [15], where a robot and a human collaborate on a physical task. Ideally, the robot wants to exhibit a behavior that conveys a clear intent to the human in the loop, so the human does not need to constantly ask the robot questions about its intentions. In this case, direct communication between the robots and the human is available, but studies on legibility impose the constraint that it is not used, preferring that the robot uses an implicit way to reveal its goals. Legible behavior results in a smoother and more natural human-robot interaction than purely functional behavior [15].

Following its introduction, the notion of legibility has been explored in several contexts, including task and motion planning, both in single-agent and multi-agent scenarios [16,30,26,27,39,38]. The main idea in this body of work is that, when the actor plans its behavior, it needs to optimize not only for achieving its goals but also for legibility, which involves choosing actions that disambiguate between the different goals that it might want to fulfill.

Legibility is only one of the concepts that arise in the context of *interpretable agent behavior*. As explained at length by Chakraborti et al. [12], if *legibility* "reduces ambiguity over possible goals that might be achieved", there are other dimensions that are equally important, namely *predictability*, which "reduces ambiguity over possible plans, given a goal/planning problem", and *explicability*, which "measures how close a plan is to the expectation of the observer, given a goal/planning problem". It is worth noting that all these three notions take the point of view of the actor, i.e. they become meaningful in the context of formulating plans that allow the actor to be legible, predictable, and explicable to the observer.

Goal recognition, on the other hand, supports the observer, which knows the set of goals that an actor might want to achieve but does not know which specific goal the actor has picked during a particular run. The observer aims to disambiguate among possible goals, identifying the one that the actor has selected based on observing the actor's actions. A goal recognition system can be thought of as an inference function that maps sequences of observed actions to goals. Goal recognition can be performed by the observer regardless of whether the actor tries to enact a legible behavior or not. Indeed, in adversarial domains, the actor might actively try to mislead the observer [29,3,7]. On the other hand, in any context in which an actor generates and implements legible plans, it is assumed that the observer runs a goal recognition system.

1.1. Pathfinding for goal legibility and goal recognition

In this paper, we focus on goal legibility and recognition in a cooperative, multi-agent *pathfinding* setting [31,33,8]. Following Bernardini et al. [8], we consider an environment with one origin and multiple destinations. A set of identical actors are available to serve requests at the destinations. An actor, after being tasked with reaching a specific destination needing attention, enters the environment at the origin and moves to that destination. An observer monitors the environment from a distance: it can see the actors' actions with accuracy (i.e. an observation corresponds to an actor's move) and wants to determine their destinations as quickly as possible after starting to observe their motion. Since, typically, the observer is tasked with monitoring several actors simultaneously, it can start watching the movement of a specific actor at any point in time, not necessarily the origin. No direct communication is available between the actors and the observer; hence, the latter can rely on its observations only to infer the actors' destinations. In this context, the goal legibility problem involves synthesizing paths for the actors that make it easy and quick for the observer to infer their destinations, and the goal recognition problem consists in formulating a function that maps sequences of observations to destinations.

In our work, we perform a *joint design* of goal legibility and goal recognition and solve both problems at the same time; namely, we build a planning system that provides the actors with legible paths and the observer with a goal recognition function associating sequences of observations with destinations. The goal of this joint design is to minimize the number of observations the observer needs to make in order to tell an actor's destination, i.e. to minimize what we call *legibility delay*.

Ideally, we would like to build a distinct path γ to each destination d. In this case, we could associate each transition in γ with d, and, consequently, as soon as the observer sees an actor making a move, it would be able to tell where the actor is going (each edge of the path corresponds to one and only one destination). Since this might not always be possible due to the topology of the environment (e.g. any two paths to two different destinations share a common edge) or budget constraints, the next best option is to reduce the overlap between paths as much as possible. Until the observer sees moves that are in common between two or more paths, it cannot disambiguate among the destinations at the end of those paths; however, it will be able to identify a specific destination after observing the first move that belongs to one path only. In consequence, the problem of producing legible behavior for the actors in a pathfinding setting with a deterministic goal recognition system boils down to building paths from the origin to the destinations that overlap as little as possible. Once this set of paths has been created, goal recognition becomes simple, reducing to a function that deterministically associates observation sequences to destinations.

Our method works in two separate phases. In an *offline* phase, a pathfinding system builds paths with minimum overlap between the origin and all the destinations and makes them available to the actors. Based on such paths, it also formulates a goal recognition function, which maps sequences of observations to destinations, and gives the observer access to it. The way the paths are built ensures that the number of observations the observer needs to make when applying the goal recognition function to identify an actor's destination is minimized regardless of when it starts monitoring the actor. At the end of this offline phase, the planner has done and ceases to communicate with the actors or the observer.

In a subsequent *online* phase, each actor entering the environment picks a destination and follows the pre-calculated path to it. The observer does not know what destination has been selected by an actor but can observe its moves and consult the observation - destination mapping. Given how the paths have been constructed, the observer is guaranteed to be able to tell the actor's destinations with certainty by making the minimum possible number of observations regardless of when it starts following the actor's movements. We assume that there is no communication between the planner, the observer and the actors during the online phase of

the operations. The actors act autonomously based on the paths previously built by the planner, and the observer decides based on the goal recognition function. The planner is not part of this phase.

1.2. Motivations

Our joint design is motivated by two types of operations. As in previous work [8], we consider *cooperative* robotic missions where the actors and the observer (a human supervisor or an automated system) do not share the same physical space and do not interact directly. The robots autonomously perform some tasks, and the observer is in charge of remotely monitoring the unfolding of the mission. The supervisor is not supposed to follow each robot's actions at all times, but, when necessary, it needs to be able to quickly assess the goal of each robot by simply looking at its behavior. For example, consider an autonomous fire-fighting system operating in a factory where some rooms are at risk of fire and need to be periodically inspected.

Our technique is useful not only in cooperative environments but also in *collaborative* ones. In this second case, the observer and the actors operate within the same space; for example, imagine an automated warehouse where a human supervisor is in charge of monitoring several robots at the same time, trying to make sense of their behavior. In these scenarios where the observer might be busy with several tasks and might have only a partial view of the environment, our system can assist it to quickly assess where each robot aims to go regardless of then it starts observing the robot and despite obstacles and other features that might impede full observability.

There might be multiple reasons why direct communication between the planner, the actors and the observer might not be possible or desirable in real-world applications. Robust, synchronous communication may not be available due to the nature of the environment; communication may not be advisable due to cybersecurity risks; or simply communication may not be practical.

In both the cooperative and collaborative scenarios outlined above, our goal is to simplify the observer's decision-making process and reduce its cognitive load. To this end, we provide the robots with a set of legible paths they can follow and the observer with a goal recognition system that maps observations with destinations. If the observer is not directly given the mapping, it could learn it by monitoring a sufficient number of runs of the systems (i.e. paths to destinations). However, this learning aspect is outside the scope of our paper, and we assume the observer is aware of the mapping either by receiving it or learning it. We give a concrete use case of our technique at the end of this section.

Example 1 *(Automated warehouse - Fig. 1)*. Consider an automated warehouse where robots enter from an origin point and must reach different destinations to service them. The robots are equivalent to one another so it does not matter which robot reaches which destination. The warehouse floor is organized into cells that the robot can traverse. Sensors can be attached to a single cell (movement sensors), or they can capture multiple cells (e.g., cameras). When a service is needed in one of the destination cells, an available robot is selected and tasked with reaching that cell and delivering the service. A pathfinding algorithm, which sits at the core of the automated warehouse, provides the robots with paths (sequences of cells to traverse) to go from the origin to the assigned destinations.

Although the warehouse is fully automated, human supervisors take turns monitoring the warehouse floor. They can be co-located with the robots (Fig. 1(a)) or they can observer them from a distance (Fig. 1(b)). They are required to verify that the operations on the warehouse floor progress smoothly and that the different locations requiring attention are adequately serviced. If not, they can bring additional robots or human operators to the floor. The supervisors are tasked with several activities, so they are not required to monitor each robot continuously, which would also be difficult when many robots are on the floor.

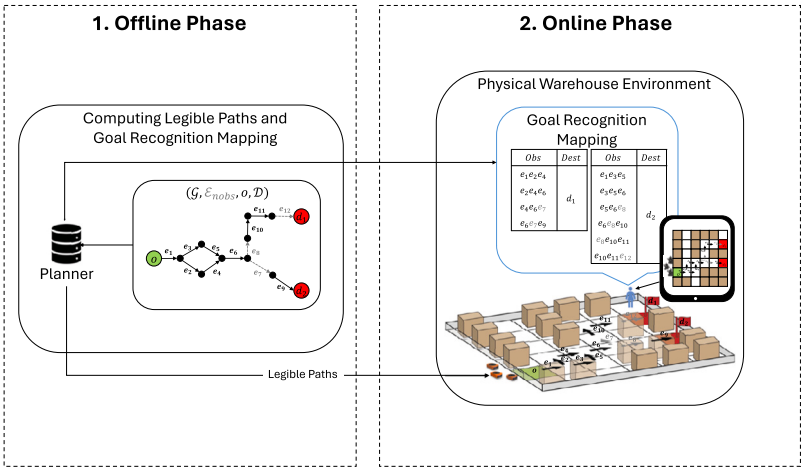
The supervisor can start observing a robot at any time (not necessarily at the time of its ingress to the floor) and can see a robot moving in the environment whenever it traverses a cell that is equipped with a sensor, but (s)he does not know the destination a robot has been assigned to. The supervisor cannot communicate directly either with the robots or with the pathfinder. In some cases, however, the supervisor needs to quickly assess the final destinations of the robots. For example, when a robot fails en route to a destination, the supervisor needs to know which destination remains uncovered to intervene. In addition, during bursts of activity at the destinations, the supervisor must decide whether to bring additional robots to the floor and at which destinations to send them.

The algorithms we propose in this paper simplify the supervisors' job as they provide the robots with legible paths and the supervisors with a simple, deterministic goal recognition system. From a practical point of view, this system can vary in its sophistication; it could be a simple look-up table or a visual interface, but - in its essence - it is a function that maps observations of snippets of paths to destinations. Hence, the supervisor will not need to perform any inference, only to consult the provided goal recognition mapping to find out with certainty where a robot's final destination.[1]

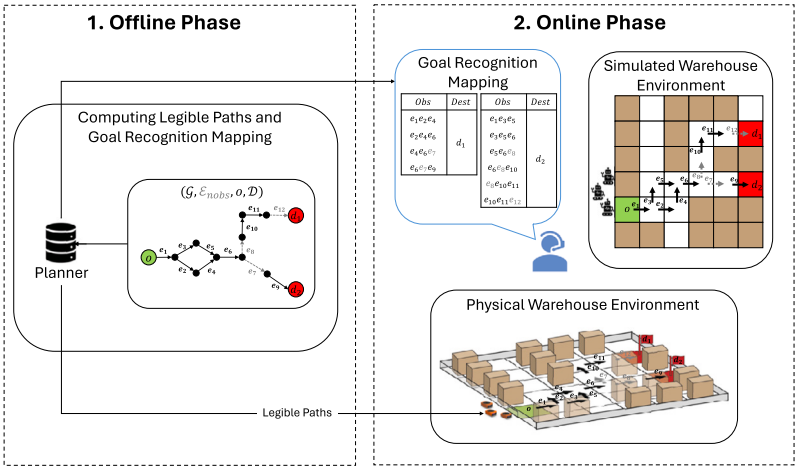
1.3. Paper outline

For synthesizing legible paths, Bernardini et al. [8] propose a reformulation approach that allows them to solve the problem by exploiting classical optimal flow techniques. In particular, the legibility problem is cast as finding the maximum flow in a suitably constructed network representing the environment. Although this approach is powerful, it is limited to *fully observable* environments. This assumption makes it unsuitable for the real-world applications we are interested in, where the observer and the actors are

[1] The study of how to visualize the goal recognition mapping is outside the scope of our work. We focus on how to solve the optimization problem of building legible paths and the corresponding function between observations and destinations, not the practical problem of presenting this function to the user.



(a)



(b)

Fig. 1. Automated Warehouse Example: (a) Example of a collaborative scenario where the actors and the observer are co-located in the same environment. (b) Example of a cooperative scenario where the observer monitors the actors from a distance, either directly or via a simulated interface.

physically separated. In these cases, various factors can hinder the observer's capacity to monitor the actors, e.g., structural obstacles or lack/failures of sensors. Hence, it is more natural to assume that the environment is only *partially observable* to the observer. In this paper, we close this gap by tackling goal legibility and recognition in a multi-agent pathfinding setting where the observer can observe the environment only partially. This extension to *partial observability* is challenging as missing observations can mislead the observer in identifying the actors' goals. The resulting theory, which has to handle partial observability, budget restrictions, and the requirement to minimize observations starting at any point (instead of the origin only) of an actor's trajectory, is much more sophisticated than the one presented by Bernardini et al. [8].

In what follows, after giving an account of related work in Section 2, we describe our setting (Section 3) and formalize the problem of pathfinding for goal legibility and recognition under partial observability (Section 4). Section 5 summarizes the approach for solving the legibility problem under full observability proposed by Bernardini et al. [8]. Subsequently (Section 6), by working through several examples, we show how a simple extension of their work to partial observability produces incorrect results, which can be avoided using a more sophisticated method. Hence, in Section 7, we lay down our full theoretical approach and, in Section 8, the related algorithms. We then present an experimental analysis showing our technique's feasibility and effectiveness (Section 9). We conclude with final considerations in Section 10. Appendix A presents an example of our construction in detail and Appendix B additional experimental results. Lists of Symbols and Abbreviations end the paper.

2. Related work

As our framework spans two well-established problems in the existing literature, this section presents a summary of prior research about the concepts of *goal legibility* and *recognition*. Additionally, we cover the intersection of our work with *goal recognition design* and flow network formalizations applied to *pathfinding*.

The main differences between our method and those present in the literature are: (a) we focus on goal legibility and recognition in a *pathfinding* setting, while most of the existing work is situated either in motion or task planning; (b) we consider a *multi-agent* scenario, while literature treats single-agent domains; (c) as we aim to endow the observer with maximal flexibility regarding when it needs to start monitoring an actor in order to identify its goal, we assume it can start *at any point*, not necessarily at the beginning of its motion, which is a common assumption in related work; and (d) we undertake a *joint design* of a goal recognition system for the observer and legible paths for the actors, while state-of-the-art methods focus on one or the other. Regarding the last point, we build legible paths for the actors and construct them in such a way as to minimize the number of observations that are needed for the observer to disambiguate between destinations. We also provide the observer with a mapping between unique sequences of observations of those paths and the corresponding destinations. Therefore, in our framework, the observer does not need to engage in probabilistic reasoning to discover an actor's destination; it is sufficient that it waits until it observes a sequence appearing in the mapping and then can tell without uncertainty the actor's destination. Although our method is deterministic (no probabilistic reasoning involved), as mentioned in the Introduction, the optimization problem we tackle is complex since partial observability, budget restrictions, and the observer's flexibility about when to start observing create a sophisticated set of constraints.

2.1. Goal legibility

In the AI and robotics communities, there has been growing interest in *interpretable agent behavior* in the past few years [17,28,22,12,48,47], stemming from the consideration that rarely, if ever, agents act in isolation from humans. Synthesizing interpretable behavior facilitates smoother Human-AI interaction and also supports *trust* in autonomy [9]. As mentioned in the Introduction, interpretability has been studied along three main dimensions, *legibility, explicability* and *predictability* [12], but, lately, some effort has been made to connect and integrate these concepts in unified frameworks [48,39]. We will limit our discussion to legibility and the most relevant related work.

Noting that humans are usually proficient at inferring the mental states underlying other agents' actions, including their goals, beliefs, desires, emotions, and thoughts, Baker et al. [6] proposed a computational framework based on Bayesian inverse planning for modeling human action understanding. The framework is based on the principle of rationality: the expectation that agents will plan approximately rationally to achieve their goals, given their beliefs about the world.

Based on Baker et al.'s study and related ones, e.g. [14], Dragan et al. [17] proposed the notion of legibility in the context of motion planning. They are interested in scenarios in which the robot and the human physically interact (e.g., they undertake a task together). The observer, which starts observing from the beginning of the interaction, is modeled as a probabilistic goal recognition system and the planning agent searches for a plan towards its true goal by favoring actions that maximize the goal's posterior probability as calculated by the observer. Fisac et al. [18], Nakahashi [41] and Bobu et al. [11] have continued research on this type of human-robot cooperation based on legibility.

Together with motion planning settings, legibility has been studied in task planning. Kulkarni et al. [26] present a deterministic methodology to address this challenge. They also introduce a second problem, *plan legibility*, wherein the actor's objective is to implicitly communicate its planned actions to an observer who is aware of its true goal. Similarly to our premise, the authors consider a partially observable setting, in which they show that partial observability can be strategically utilized to optimize either legibility or obfuscation. Further, Kulkarni et al. [27] address both legibility and obfuscation as a unified problem framed as a *mixed-observer controlled observability planning problem* (MO-COPP). The authors propose two different approaches to solving MO-COPP: one formulating it as a constraint optimization problem and the other as a heuristic search problem. In their approach, the actor exploits partial observability to simultaneously reveal and conceal its plan towards a cooperative observer and an adversarial one, respectively. The actor leverages its knowledge of both the observers' perception capabilities and their prior observations and generates different effects in the mental models of the cooperative and adversarial observer by choosing a single observation sequence. Kulkarni et al.'s research diverges from ours in several ways; it concentrates on a single-actor offline task-planning setting, where the observer exclusively receives observations after the agent has completed its execution of a plan. In contrast, our approach accounts for multiple actors in a pathfinding setting and the observer monitors them while they are actively moving towards their targets.

Another approach to legibility for task planning is proposed by Miura et al. [38], who provide an extension to Dragan et al.'s formulations for legible planning to stochastic environments modeled as Markov Decision Processes (MDPs). They define the concept of *legible MDP*, whose optimal policy conveys the agent's intention as much as possible. This is obtained by extending the definition of MDPs by making the reward depend on the beliefs of the observer. The observer is, in turn, assumed to update its beliefs based on Bayesian reasoning. The authors illustrate through examples that their approach is capable of maximizing legibility in stochastic environments and that using legibility as an objective improves the interpretability of agent behavior in several scenarios.

The notion of goal legibility is also relevant in other domains related to planning. Chandra et al. [13], for example, investigate this concept for effective storytelling. They suggest that, just as Baker et al. [6] cast "action understanding as inverse planning", acting can be seen as "inverse inverse planning", i.e. the task of choosing actions to manipulate an inverse planner's inferences. Their goal is to optimize animations in computer graphics with respect to simulated audience members' experiences.

2.2. Goal recognition

Beyond addressing the legibility problem inherent in the actor's viewpoint within a human-in-the-loop scenario, our framework also tackles the interpretability challenge arising from the observer's perspective, denoted as goal recognition. Goal recognition is the

task of inferring an agent's goal, from a set of hypotheses, given a model of the environment dynamic, and a sequence of observations of such agent's behavior [35].

Across the scientific literature, the scope of applications of goal recognition, more broadly referred to as plan recognition, extends across various fields. Note that goal recognition is essentially a sub-problem of plan recognition. Notable instances include its use in interactive conversational systems for identifying a user's broader domain goals [10,40], video games to determine a player's objectives [36,37], machine translation to generate a user's intentional structure and dialogue history [1]. It also finds application in cyber and physical security to prevent an adversarial agent from reaching its target [3,33], military operations to support decision-making processes [23], and healthcare, where it is focused on activity recognition [5,46].

Given this extensive range of applications, each with its own specificities, an array of various methods have been employed to tackle plan recognition under both full and partial observability. These methods include but are not limited to machine learning and Bayesian networks [23,10,53,40,21,36,37,55], heuristic classical search and graph theory[43,42,24,3], and, in earlier research, rule-based methodologies [52]. Plan recognition problems vary based on the observed agent's awareness of being monitored, leading to different problem formalizations such as decision-theoretic, game-theoretic, and probabilistic, among others. These include *intended plan recognition* with a cooperative agent [10], *keyhole plan recognition*, where an agent is unaware of being monitored [21,33], and *adversarial plan recognition*, where the agent actively aims to deceive the observer [29,3]. Work on *deception* has received considerable attention lately, with techniques for deceptive path planning [32], UAV maneuvering [45] and social communication [2], among others.

Early research on goal recognition relies on expert knowledge of the domain in which the agent is acting and consists in generating a plan library enclosing all its possible plans. However, this is not a practical approach in real-world scenarios [21]. Subsequently, Ramirez and Geffner [43,44] pioneered a shift in perspective, bridging the gap between plan recognition and automated planning. By exploiting the generality of classical AI planning, the authors propose a novel paradigm that regards plan recognition as "planning in reverse". Their planning formalization relies on abductive reasoning, which involves making assumptions to explain observed data. The main insight they build upon is that the probability of a plan can be linked directly to its cost. An agent is assumed to take an optimal [43] or least sub-optimal [44] path to the goal. In the latter work, the authors generate a probability distribution over the possible actor's goals based on the cost difference between the cheapest available plan that conforms to observations and the cheapest plan that does not. Expanding upon this concept, there is a growing body of research applying it to diverse settings and addressing various challenges (see Meneguzzi and Pereira's survey [35]).

Building on Ramirez and Geffner's work, Masters and Sardina's research [31,33] focuses on goal recognition in navigational domains. We share this focus with them and also the underlying motivations. Navigational domains underpin many significant applications of goal recognition and legibility and understanding algorithms in path planning often works as a precursor for developing similar algorithms in task planning. However, differently from them, we tackle both perspectives of the recognition problem, the observer's and the actors' ones, while they address the observer's perspective only. Masters and Sardina import Ramirez and Geffner's account into the realm of path planning and show that a probability distribution that ranks candidate goals in the same order as Ramirez and Geffner can be obtained purely based on the actor's starting and current locations without considering any observations between these points. They also demonstrate that a *Radius of Maximum Probability* (the distance from a goal within which that goal is guaranteed to be the most probable) can be calculated from relative cost-distances between the candidate goals and a start location, without needing to calculate any actual probabilities [33]. Finally, they generalize their framework to continuous domains.

In subsequent work [34], Masters and Sardina question the assumption of rationality that is implicit in contemporary work on cost-based goal-recognition, i.e. that observed agent behavior is more or less optimal. Motivated by real-world applications in which agent rationality cannot be taken for granted, they provide a definition of rationality appropriate to situations where the ground truth is unknown, define a rationality measure that quantifies an agent's expected degree of sub-optimality, and define a self-modulating probability distribution formula for goal recognition. They demonstrate the viability of their approach in path-planning domains.

Kulkarni et al. [27] present the development of a goal recognition system based on a Mixed-Observer Controlled Observability Planning Problem (MO-COPP) approach. Specifically, they consider the problem where an autonomous agent needs to act in a manner that clarifies its objectives to cooperative agents while simultaneously preventing adversarial agents from inferring those objectives. They provide two solution approaches: one provides an optimal solution to the problem given a fixed time horizon by using an integer programming solver, the other provides a satisficing solution using heuristic-guided forward search to achieve prespecified amount of obfuscation and legibility for adversarial and cooperative agents respectively.

The topic of learning to cooperate with friends and compete with foes has been tackled also in the context of multi-agent reinforcement learning, with work by Strouse at al. [50]. They take an alternative view with respect to related work as their technique does not require an explicit model of the other agent. The approach they propose is based on an information theoretic formulation of the problem of sharing and hiding intentions and can be combined with scalable policy-gradient methods commonly used in deep reinforcement learning.

2.3. Goal recognition design

Our technique bears some similarities with the work of Keren et al. [25] on Goal Recognition Design (*GRD*). In particular, the notion of worst-case distinctiveness (*wcd*), which is used in GRD, is connected with our concept of legibility delay (see Definition 5). A *wcd* equal to k indicates that k is the maximal prefix length of any path an agent may take to reach its goal before it becomes clear to the observer. This means that there are at least two paths that share a prefix of length k and go to two different destinations. On the other hand, a legibility delay equal to j indicates that, given any two paths from the origin to two different destinations, they

do not share any subpath of length j. It follows from the definitions of *wcd* and legibility delay that the legibility delay is always greater or equal to the *wcd* plus one (because the legibility delay counts up to the transition at which the goal becomes clear to the observer, while the *wcd* counts up to the transition *before* the goal becomes clear to the observer). Since the *wcd* is calculated over prefixes of plans only, if two paths have an overlapping fragment that is not a prefix, the *wcd* is zero, while the legibility delay is the length of the fragment plus one. Despite the connection between *wcd* and legibility delay, the two concepts are used in different ways in GRD and in our framework. GRD aims at modifying the environment to facilitate (or hinder) legibility, while we do not alter the environment; we construct legible paths in it.

The concepts of *GRD* and *wcd* have been extended to *probabilistic* settings (*Stochastic-GRD*), i.e. settings in which the agent's actions are non-deterministic [51]. Another proposed extension is an *online* setting (*Active-GRD*), where both the observer's and the agent's actions are interleaved [20]. Recently, Au [4] proposed a novel framework termed *Extended GRD*, providing solutions for scenarios where an agent pursues multiple goals instead of just one goal. These three extensions would be interesting areas of further research when considering our concept of legibility delay.

2.4. Pathfinding

We define legibility in the context of pathfinding, which, given its importance in several practical applications, has been studied for many years. We refer the readers to recent survey papers on this topic for an overview of the main methods [49]. Here, we focus on the connection between our work and Yu and La Valle's algorithm for solving anonymous multi-agent pathfinding problems [54]. Although we take inspiration from their work in expressing our problem of finding legible paths in terms of calculating the maximum flow of a suitably constructed network, our settings are different. Time is central to Yu and La Valle's formulation. They work on a time-expanded network, assume global synchronization between the agents, which start at the same time and move in a coordinated fashion, and their goal is to minimize makespan or flowtime. On the other hand, we do not represent time explicitly. We do not impose synchronization constraints among the agents (as well as the observer) and our optimization functions involve the legibility delay and path costs. It is conceivable to use time as cost, and we will explore that in future work.

3. Setting

3.1. Model of the environment

We model the environment as a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, representing different locations in the environment, and $\mathcal{E}$ is the set of edges, representing connections between the locations. We work with *multigraphs* because the constructions in the technical sections below become simpler when using this concept, which is more flexible than the notion of graph. However, in the examples, when there are no parallel edges, we depict graphs with edges represented as ordered pairs of vertices.

We assume the existence of two functions $\theta : \mathcal{E} \to \mathcal{V}$ and $\kappa : \mathcal{E} \to \mathcal{V}$ with the interpretation that $\theta(e)$ and $\kappa(e)$ represent the tail and, respectively, the head of an edge $e \in \mathcal{E}$. Given a vertex $v \in \mathcal{V}$, we define $in(v)$ and $out(v)$ as follows:

$$in(v) = \{e \in \mathcal{E} \mid \kappa(e) = v\}, \quad out(v) = \{e \in \mathcal{E} \mid \theta(e) = v\}$$

A *walk*[2] from a node v and a node w is a sequence of edges $\gamma = (e_1, \dots, e_l)$ such that $\theta(e_1) = v$, $\kappa(e_l) = w$, and $\theta(e_h) = \kappa(e_{h-1})$ for all $h = 2, \dots, l$. We indicate the length of a walk γ as $l_\gamma = l$. In the examples, when we represent a walk explicitly by indicating its edges, we omit the commas between them for brevity.

A node $o \in \mathcal{V}$ is designated as the *origin* and a set $\mathcal{D} \subseteq \mathcal{V}$ as the possible *destinations*, with $|\mathcal{D}| > 1$. We assume that o has only outgoing edges ($in(o) = \emptyset$) and the nodes $d \in \mathcal{D}$ have only incoming edges ($\forall d \in \mathcal{D},\ out(d) = \emptyset$).

3.2. Model of the actors

The environment is populated by multiple, identical *actors* that move in it. We model their movement as they traverse the graph edges.

The actors enter the environment and start to move from the origin o. When getting into the environment, each actor is tasked with reaching a destination in $\mathcal{D}$. Since the actors are equivalent to one another, it does not matter which actor reaches which destination.

At the core of our approach is the construction of a set of walks $\mathcal{P}$ that the actors can use to reach the destinations $\mathcal{D}$ from the origin o.

3.3. Model of the observer

An *observer* monitors the environment from a distance. The observer can detect the actors traversing the graph edges but not all of them. More specifically, the set of edges $\mathcal{E}$ is partitioned into two subsets, $\mathcal{E} = \mathcal{E}_{obs} \cup \mathcal{E}_{nobs}$, where $\mathcal{E}_{obs}$ and $\mathcal{E}_{nobs}$ are respectively

[2] The introduction talks about *paths* following pathfinding literature, but the precise term in graph theory is *walks*, which we will use throughout the paper.

observable and *unobservable* edges. The observer can detect an actor traversing an observable edge but not an unobservable one. We assume that the actors' edge traversal (not their permanence on nodes) generates observations for the observer and that these observations, when possible on the edges $\mathcal{E}_{obs}$, are accurate. Hence, we do not consider noisy sensors in this work.

More formally, considering a set of observation tokens $\mathcal{O} = \mathcal{E}_{obs} \cup \{\text{null}\}$, we define an *observation model* for the observer as a function $\iota : \mathcal{E} \to \mathcal{O}$ that associates tokens $o \in \mathcal{O}$ with edges $e \in \mathcal{E}$, as follows:

$$\iota(e) = \begin{cases} e & \text{if } e \in \mathcal{E}_{obs} \\ \text{null} & \text{if } e \in \mathcal{E}_{nobs} \end{cases} \tag{1}$$

A walk $\gamma = (e_1, \dots, e_l)$ generates a sequence of observations $(\iota(e_1), \dots, \iota(e_l))$, which we indicate as $\iota(\gamma)$ with a slight abuse of notation.

The observer is aware of the possible destinations the actors might want to achieve, hence it knows the set D, but it does not know which destination an actor needs to reach when it enters the environment. The goal of the observer is to infer that based on the observations generated by the actor's traversal of the edges in $\mathcal{E}_{obs}$. More formally, the observer runs a *goal recognition system* that can be formalized as follows.

Consider all walks in $\mathcal{G}$ of a given length $s \in \mathbb{N}^+$:

$$\mathcal{Q}^{(s)} = \{(e_1, \dots, e_s) \in \mathcal{E}^s \mid \theta(e_h) = \kappa(e_{h-1})\ h = 2, .., s\}$$

and the corresponding set of all observation sequences generated by $\mathcal{Q}^{(s)}$:

$$\mathcal{I}^{(s)} = \{(\iota(e_1), \dots, \iota(e_s)) \mid (e_1, \dots, e_s) \in \mathcal{Q}^{(s)}\}$$

The observer's goal recognition system consists of choosing a value s and constructing a function $\Omega^{(s)}$ as follows:

$$\Omega^{(s)} : \mathcal{I}^{(s)} \to D$$

The function $\Omega^{(s)}$ associates sequences of observations of a given length s with destinations. The idea is that the observer, by applying $\Omega^{(s)}$ to the observations generated by a walk γ of length s, is able to tell the destination of an actor that is following a walk γ', which contains the subwalk γ, to reach such a destination.

Note that, given a walk from o to a destination d, $\gamma = (o, e_1, \dots, d)$, the observer can start observing an actor following γ at any edge $e_i \in \gamma$, not necessarily the origin o. In addition, the observer's goal is to tell the actor's final destination as early as possible after starting observing its movement. Hence, given a sequence of observations $(\iota(e_1), \dots, \iota(e_s))$, generated by the walk fragment $(e_1, \dots, e_s)$, we want to make s as small as possible for the observer, while enabling it to infer the correct destination of the actor.

3.4. Legibility problem under partial observability and solution

We call the tuple $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$ an instance of the *legibility problem under partial observability* (PO-LP). When $\mathcal{E}_{nobs} = \emptyset$, the tuple $(\mathcal{G}, o, D)$ is an instance of the legibility problem under full observability (LP) as defined by Bernardini et al. [8].

A *solution* of a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$ is a pair $(\mathcal{P}, \Omega^{(s)})$ such that, for each walk $\gamma \in \mathcal{P}$ leading to a destination $d \in D$, and for every subwalk γ' of γ of length s, we have that:

$$\Omega^{(s)}(\iota(\gamma')) = d$$

As we will see in the next section, in this paper, we focus on finding solutions that minimize the length s or that trade s against the complexity of the set of walks $\mathcal{P}$.

4. Problem statement

Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$, our goal is twofold. We want to provide the actors that enter the environment with walks to the destinations that they have been assigned to serve. At the same time, we want to enable the observer to recognize the actors' destinations as soon as possible after it starts observing the actors' movements in the environment. As mentioned in the Introduction, we satisfy those goals by proposing a pathfinding algorithm that finds a set of walks, one to each destination, that present minimal overlap between each other. The more distinct the walks are, the easier is for the observer to associate them to their respective destinations. In what follows, we formalize these ideas by linking the concept of legibility with that of walk overlap and present the optimization problems that we will tackle.

4.1. Formalization of legibility

We now introduce some preliminary notation to be able to define the concept of legibility formally. We start with the idea that we want to find a set of walks going from the origin to each destination.

Given a set of walks $\mathcal{P}$ from o to D, we call $\mathcal{P}(d)$ the subset of walks in $\mathcal{P}$ that connect o to destination d.

Definition 1. We say that a set of walks $\mathcal{P}$ is (o, D)-*connecting* if, for every $d \in D$, there exists a unique walk $\gamma \in \mathcal{P}$ from o to d (i.e. $\forall d \in D, |\mathcal{P}(d)| = 1$).

We introduce an equivalence relation on the set of walks $\mathcal{Q}^{(s)}$ based on the following observation. The edges in $\mathcal{E}_{nobs}$ cannot be distinguished by the observer and, consequently, two walks that differ only for their unobservable edges give rise to the same sequence of observations for the observer. Hence, they are considered equivalent from a legibility point of view.

Definition 2. Given $\gamma' = (e'_1, \dots, e'_s)$ and $\gamma'' = (e''_1, \dots, e''_s)$ in $\mathcal{Q}^{(s)}$, we say that the two subwalks are *observably equivalent* if $\iota(\gamma') = \iota(\gamma'')$ and write

$$(e'_1, \dots, e'_s) =_{obs} (e''_1, \dots, e''_s)$$

The definition implies that, for every $i = 1, \dots, s$, either $e'_i = e''_i \in \mathcal{E}_{obs}$, or $e'_i, e''_i \in \mathcal{E}_{nobs}$ (and are possibly different).

We are now ready to formalize the concept of legibility under partial observability. We intend the legibility of a set of walks not as a binary concept (legible, not legible), but as a spectrum. More specifically, we attach a number to the legibility of the set that indicates how many observations the observer needs to make in order to find out the destination of an observed actor, which follows one of the walks in the set.

Definition 3. Given $s \in \mathbb{N}^+$, a walk γ in $\mathcal{G}$ is said to be *s-observable* if, in every subwalk of length s of γ, there is always at least one observable edge.

Definition 4. Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ is called *s-legible*, with $s \in \mathbb{N}^+$, if

(i) every $\gamma \in \mathcal{P}$ is s-observable;
(ii) $\forall\ d, d' \in \mathcal{D}$ with $d \neq d'$, $\forall\ \gamma \in \mathcal{P}(d)$ and $\forall\ \gamma' \in \mathcal{P}(d')$, we have that $(\gamma_{x+1}, \dots, \gamma_{x+s}) \neq_{obs} (\gamma'_{x'+1}, \dots, \gamma'_{x'+s})$, $\forall x, x' \in \mathbb{N}$ such that $x + s \leq l_\gamma$ and $x' + s \leq l_{\gamma'}$.

Definition 4 tells us that a set of $(o, \mathcal{D})$-connecting walks $\mathcal{P}$ is s-legible if (i) no walk in $\mathcal{P}$ has s unobservable consecutive edges, and (ii) given any two walks from the origin to two different destinations, they do not share any observably equivalent subwalk of length s (hence, they give rise to two different sequences of observations). If condition (i) is not satisfied, the observer would not see anything in s steps, so it would not be able to decide where the actor is going; if condition (ii) is not satisfied, the observer would see a walk fragment of s steps compatible with multiple destinations, so again it would not be able to make a decision. Note that condition (ii) does not subsume condition (i). We would get a different definition of s-legibility by removing condition (i) as we would accept the presence in $\mathcal{P}$ of subwalks of length s composed of unobservable edges only. We want to avoid this possibility for two reasons. From a philosophical point of view, it seems counter-intuitive to talk about s-legibility for a set of walks when they present segments of s unobservable nodes as the observer would not receive any useful information when the actor traverses them. From a technical point of view, as we will see in Section 7, we rely on this assumption in the construction of the auxiliary graphs that we use to solve the problems that involve s-legibility. Condition (i) allows us to obtain a bijection between walks in the original graph $\mathcal{G}$ and walks in these auxiliary graphs.

We can formally tie the concept of legibility to that of the solution of a PO-LP instance, as the following result shows.

Proposition 1. *Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, suppose that $\mathcal{P}$ is an $(o, \mathcal{D})$-connecting set of walks that is s-legible. Then, there exists $\Omega^{(s)} : \mathcal{I}^{(s)} \to \mathcal{D}$ such that $(\mathcal{P}, \Omega^{(s)})$ is a solution of the PO-LP instance.*

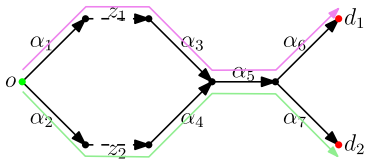
Proof. We define $\mathcal{I}^{(s)}_{\mathcal{P}} \subseteq \mathcal{I}^{(s)}$ as the set of observations generated by subwalks of walks in $\mathcal{P}$ of length s. For every $(\iota_1, \dots, \iota_s) \in \mathcal{I}^{(s)}_{\mathcal{P}}$, we consider a walk γ in $\mathcal{P}$ and an index j such that $\iota(\gamma_{j+1}, \dots, \gamma_{j+s}) = (\iota_1, \dots, \iota_s)$ and we define $\Omega^{(s)}(\iota_1, \dots, \iota_s) = d$, where d is the destination of γ. Thanks to Definition 4, this definition of $\Omega^{(s)}$ on $\iota^{(s)}_{\mathcal{P}}$ is well posed. Finally, we define arbitrarily $\Omega^{(s)}$ on the remaining subset $\iota^{(s)} \setminus \iota^{(s)}_{\mathcal{P}}$. By construction $(\mathcal{P}, \Omega^{(s)})$ is a solution of the PO-LP instance. □

From now on, we will focus on the analysis of s-legible sets, which are, on the basis of the result above, the core element of our approach.

Notice that the concept of s-legibility under partial observability coincides with the concept of s-legibility as introduced in Bernardini et al. [8] if there are no unobservable edges. In addition, given point (i) in Definition 4, the concept of 1-legibility for a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ is equivalent to the concept of 1-legibility for the corresponding LP instance $(\mathcal{G}, o, \mathcal{D})$. This paper focuses on s-legibility for PO-LP instances with $s > 1$, which is a new problem that cannot be solved via a straightforward reduction to the full observability case.

It is intuitive to see that if a set of walks is s-legible, it will also be $(s + j)$-legible for any $j \in \mathbb{N}^+$. Since we want to allow the observer to infer the actors' destinations as soon as possible after it starts observing them, we are interested in making s as small as possible. Hence, we have the following definition.

Definition 5. Given an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$, the minimum s such that $\mathcal{P}$ is s-legible is called the *legibility delay* of $\mathcal{P}$ and denoted as $LD(\mathcal{P})$.



(a)

Obs	Dest
$(\alpha_1\ null)$	
$(null\ \alpha_3)$	
$(\alpha_3\ \alpha_5)$	d_1
$(\alpha_5\ \alpha_6)$	

Obs	Dest
$(\alpha_2\ null)$	
$(null\ \alpha_4)$	
$(\alpha_4\ \alpha_5)$	d_2
$(\alpha_5\ \alpha_7)$	

(b)

Fig. 2. (a) Graph $\mathcal{G}$ and 2-legible $(o, \mathcal{D})$-connecting walks $\mathcal{P} = \{\gamma_1, \gamma_2\}$, with γ_1 depicted in violet and γ_2 depicted in green. (b) Goal recognition function $\Omega^{(2)}$ that associates each sequence of observations of length 2 with its corresponding destination. The observer, by consulting this mapping, can tell where the actor is going after making two observations. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, we can construct many different $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$. Our goal is to find the set $\mathcal{P}$ that is characterized by the minimum legibility delay, as indicated by the following definition.

Definition 6. A PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ is called *s-legible* if there exists an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ that is s-legible. The *legibility delay* of $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, denoted it as $LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, is the minimum legibility delay among all possible $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$. In formula,

$$LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D}) = \min_{\mathcal{P}\,(o,\mathcal{D})-conn.} LD(\mathcal{P})$$

Based on the definitions above, we study s-legibility by stating three optimization problems, as indicated in the following section.

Example 2 *(2-legibility).* Consider the graph $\mathcal{G}$ in Fig. 2(a), where o is the origin and d_1 and d_2 are the destinations that the actors need to service. The observer can see the actors traversing all edges except for z_1 and z_2, which are unobservable. In this figure and the following ones, unobservable edges are represented by using dotted lines.

Recall that our goal is twofold. On the one hand, we want to provide the actors with walks to reach d_1 and d_2 from o and, on the other hand, we want to facilitate the observer in finding out where an actor that traverses these walks aims to go. Looking at $\mathcal{G}$, we can see that observing α_5 creates ambiguity for the observer as any walk to d_1 and d_2 must go through it. The unobservable edges z_1 and z_2 also create ambiguity because the observer does not obtain any information when the actors traverse them.

Assume that we choose the set of $(o, \mathcal{D})$-connecting walks $\mathcal{P} = \{\gamma_1, \gamma_2\}$ to reach destinations d_1 and d_2 (γ_1 is depicted in violet and γ_2 in green). First note that both γ_1 and γ_2 are 2-observable because, in every subwalk of length 2 of γ_1 and γ_2, there is always at least one observable edge. However, they are not 1-observable due to the presence of the unobservable edges z_1 and z_2. Consequently, the two walks are not 1-legible as they do not satisfy condition (i) of Definition 4. They are instead 2-legible, meeting both conditions (i) and (ii) of Definition 4. Given any walk fragment of length 2, there is always at least one observable edge, and, γ_1 and γ_2 do not share any observably equivalent subwalk of length 2. This means that any walk fragment of length 2 gives rise to a different observation sequence, which allows the observer to disambiguate between d_1 and d_2.

Based on γ_1 and γ_2, we can construct a function $\Omega^{(2)}$ for the observer that associates observation sequences of length 2 with destinations as indicated in Fig. 2(b). The observer, by making two observations, can find out the actors' destinations with certainty. For example, if the observer starts monitoring an actor that follows the violet path when it is traversing z_1, it will have to make another observation, namely α_3, before being able to tell that the actor is going to d_1 because z_1 produces the *null* observation. Similarly, if the observer starts monitoring an actor that follows the green path from α_5, it will have to make another observation, namely α_7, before being able to tell that the actor is going to d_2 because α_5 is an edge in both γ_1 and γ_2 that does not allow the observer to disambiguate between the two. Note that the mapping in Fig. 2(b) gives the observer the freedom to start observing the actors at any time during their movement, not necessarily at the origin.

Can we do better than having a legibility delay of 2 for this problem? Can we disambiguate between d_1 and d_2 by making one instead of two observations? In some cases, this is possible, e.g. if the observer sees α_6 or α_7. However, our definitions focus on the worst-case scenario for the observer and want to provide a number for the legibility delay that works in every situation. As noted above, if one observation is sufficient when the observer sees α_6 or α_7, this is not the case when the observer sees a *null* token (when the actor traverses z_1 and z_2) or α_5, which belongs to both γ_1 and γ_2. Those observations would not help the observer. Two observations are the maximum number of observations required to disambiguate between d_1 and d_2 in every situation.

Example 3 *(Violation of 2-legibility).* Let us consider the different graphs represented in Fig. 3 and a set of $(o, \mathcal{D})$-connecting walks $\mathcal{P} = \{\gamma_1, \gamma_2\}$ (we follow the same visual conventions as in the previous example). Let us study whether the two walks γ_1 and γ_2 are 2-legible in the various cases. Recall that two walks are not 2-legible if they have two consecutive unobservable edges (see case 1) in Fig. 3), or if they share observably equivalent subwalks of length two. The second condition includes several cases, and, in particular, the two walks are not 2-legible if they share: a) two consecutive observable edges (see case 2)); b) one observable edge followed by one unobservable edge (case 3)) or c) vice versa (case 4)); d) one observable edge that bifurcates into two different unobservable edges (case 5)); and e) two different unobservable edges that merge into one observable edge (case 6)). In all those cases, we can find an ambiguous subwalk of length 2 that does not allow the observer to establish where an actor that traverses that subwalk aims

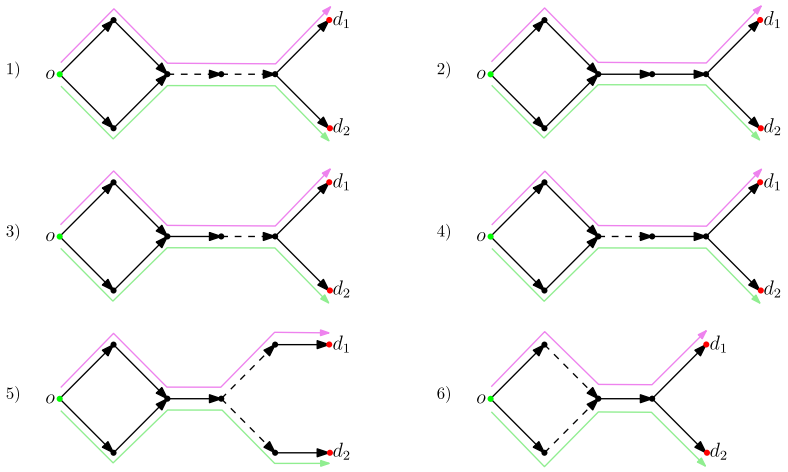


Fig. 3. Examples of two walks, γ_1 depicted in violet and γ_2 depicted in green, that are not 2-legible.

to go. Let us take the last case, case ii-e, as an example. Regardless of whether the actor is traversing the violet or green walks, if the observer starts observing at time step 2, it will receive a null token for the traversal of one of the unobservable edges and then an observation relating to the shared visible edge. Hence, the two sequences of observations would be the same, which would not allow the observer to tell if the actor is going to d_1 or d_2.

4.2. Optimization problems

Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, the first optimization problem that we tackle revolves around finding the legibility delay of an instance, i.e. the minimum legibility delay among all possible $(o, \mathcal{D})$-connecting set of walks that can be constructed in $\mathcal{G}$. We also want to produce a set of walks that exhibits that delay.

Problem 1. Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, determine its legibility delay $LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ and find an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ such that $LD(\mathcal{P}) = LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$.

The following remark explains how we treat those destinations $d \in \mathcal{D}$ reachable from o in a number of steps smaller than s when checking the s-legibility.

Remark 1. In checking the s-legibility of a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$, those destinations $d \in \mathcal{D}$ reachable from o in a number of steps smaller than s do not play any role. Precisely, define

$$\mathcal{D}_{<s} = \{d_i \mid \exists\, \gamma \text{ walk in } \mathcal{G} \text{ from } o \text{ to } d_i,\ l_\gamma \leq s-1\} \tag{2}$$

and let $\mathcal{P}_{<s}$ be any $(o, \mathcal{D}_{<s})$-connecting set of walks in $\mathcal{G}$ of length at most $s-1$. Then, if $\mathcal{P}$ is $(o, \mathcal{D} \setminus \mathcal{D}_{<s})$-connecting and s-legible, then, $\mathcal{P} \cup \mathcal{P}_{<s}$ is $(o, \mathcal{D})$-connecting and s-legible. This is because the walks in $\mathcal{P}^{(s)}$ do not contain any subwalk of length s. In particular,

$$LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D}) = LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D} \setminus \mathcal{D}_{<s})$$

We now introduce a *cost* on walks in the following way (similar to Bernardini et al. [8]'s work). Given a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we introduce a weight vector $W \in (\mathbb{R}^+)^{\mathcal{E}}$. The weight of a walk $\gamma = (e_1, \ldots, e_l)$ is defined as

$$W(\gamma) = \sum_{h=1}^{l} W_{e_h} \tag{3}$$

Finally, the cost of a set of walks $\mathcal{P}$ is defined as follows:

$$C(\mathcal{P}) = \sum_{\gamma \in \mathcal{P}} W(\gamma) \tag{4}$$

We can now enunciate two optimization problems that investigate the trade-off between cost and legibility under partial observability.

Problem 2. Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ and an admissible legibility delay s, find an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ that is at the most s-legible and minimizes the cost $C(\mathcal{P})$. Formally,

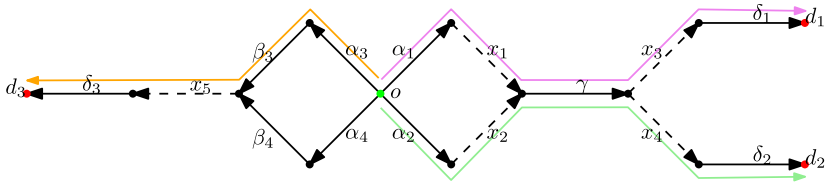


Fig. 4. Graph $\mathcal{G}$ and three (o, D)-connecting walks depicted in violet, green and orange.

$$C^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,D)}(s) = \min_{\substack{\mathcal{P}(o,D)-conn. \\ LD(\mathcal{P})\leq s}} C(\mathcal{P})$$

We put $C^{min}(s) = \infty$ if $LD(\mathcal{G}, \mathcal{E}_{nobs}, o, D) > s$.

Problem 3. Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$ and an available budget B, find an (o, D)-connecting set of walks $\mathcal{P}$ that has cost $C(\mathcal{P}) \leq B$ and minimizes the legibility delay. Formally,

$$LD^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,D)}(B) = \min_{\substack{\mathcal{P}(o,D)-conn. \\ C(\mathcal{P})\leq B}} LD(\mathcal{P})$$

We now propose an example to consider the problems stated above within a practical case.

Example 4. Consider graph $\mathcal{G}$ represented in Fig. 4, where $D = \{d_1, d_2, d_3\}$ and $\mathcal{E}_{nobs} = \{x_1, x_2, x_3, x_4, x_5\}$.

First, notice that the corresponding LP instance $(\mathcal{G}, o, D)$ where all edges are observable is 2-legible and a possible (o, D)-connecting 2-legible set of walks $\mathcal{P}$ is as follows:

$$\mathcal{P} = \{(\alpha_1 x_1 \gamma x_3 \delta_1), (\alpha_2 x_2 \gamma x_4 \delta_2), (\alpha_3 \beta_3 x_5 \delta_3)\}$$

In the full observability case, the goal recognition function run by the observer is as follows:

Subwalks	Dest
$(\alpha_1 x_1)$	d_1
$(x_1\ \gamma)$	
$(\gamma\ x_3)$	
$(x_3 \delta_1)$	

Subwalks	Dest
$(\alpha_2 x_2)$	d_2
$(x_2\ \gamma)$	
$(\gamma\ x_4)$	
$(x_4 \delta_2)$	

Subwalks	Dest
$(\alpha_3 \beta_3)$	d_3
$(\beta_3 x_5)$	
$(x_5 \delta_3)$	

However, the set is not 2-legible with respect to the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$, as the two first walks present subwalks of length 2 that are observably equivalent, respectively $(x_1\gamma)$ and $(x_2\gamma)$, because x_1 and x_2 are unobservable. Both subwalks result in the same observation sequence $(null, \gamma)$. An observer seeing an actor traversing the subwalk $(x_1\gamma)$ or $(x_2\gamma)$ would not be able to determine where the actor aims to go. More generally, note that an (o, D)-connecting 2-legible set of walks for the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$ cannot exist as any pair of walks to d_1 and d_2 have to contain the subwalks, respectively, $(x_1\gamma)$ and $(x_2\gamma)$.

Similarly, the set $\mathcal{P}$ does not have legibility 3. Note that the two subwalks $(x_1\gamma x_3)$ and $(x_2\gamma x_4)$ are observably equivalent and any pair of walks to d_1 and d_2 have to contain those subwalks. Hence, an (o, D)-connecting 3-legible set of walks for the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$ cannot exist.

The correct legibility of the PO-LP instance is $\mathcal{P}$ 4, and $\mathcal{P}$ is a possible (o, D)-connecting 4-legible set of walks, with goal recognition function run by the observer is as follows:

Obs	Dest
$(\alpha_1$ null γ null$)$	d_1
(null γ null δ_1)	

Obs	Dest
$(\alpha_2$ null γ null$)$	d_2
(null γ null δ_2)	

Obs	Dest
$(\alpha_3\beta_3$ null $\delta_3)$	d_3

5. Background: full observability

In this section, we give a brief account of the theoretical approach for solving the legibility problem under full observability proposed by Bernardini et al. [8]. They show that the three problems stated in Section 4 for an instance of the legibility problem under full observability can be reformulated in terms of finding suitable flows in a network that can be constructed based on the instance. After summarizing their reformulation approach, we will show in the following sections how their method can be generalized to tackle partial observability.

5.1. 1-Legibility as a network flow problem

Below, we recall the basic results for the case of 1-legibility as first presented in Bernardini et al. [8], which will play a pivotal role also in the new context of partial observability. The key idea is to transform the legibility problem into a flow network problem and then apply efficient classical tools such as Ford-Fulkerson [19]'s algorithm.

Let us start with considering Problem 1 for the case of 1-legibility.

Consider an LP instance $(\mathcal{G}, o, \mathcal{D})$ and make a simple modification to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by creating a new fictitious destination $\bar{d}$, which we add to $\mathcal{V}$, and generating an edge from each destination $d \in \mathcal{D}$ to $\bar{d}$, which we add to $\mathcal{E}$. This new graph $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ is called the *modified graph*. We equip this graph with a capacity function $u : \bar{\mathcal{E}} \to \mathbb{R}^+$ such that $u(e) = 1$ for every $e \in \bar{\mathcal{E}}$. A *feasible o-$\bar{d}$ flow* on $\bar{\mathcal{G}}$ of value $\nu(f)$ is any function $f : \bar{\mathcal{E}} \to \mathbb{R}^+$ such that $f(e) \le u(e) = 1$ for every $e \in \bar{\mathcal{E}}$ and the following conservation laws are satisfied:

$$\sum_{e:\theta(e)=v} f_e = \sum_{e:\kappa(e)=v} f_e \; \forall v \in \bar{\mathcal{V}} \setminus \{o, \bar{d}\}, \qquad \sum_{e:\theta(e)=o} f_e = \sum_{e:\kappa(e)=\bar{d}} f_e = \nu(f)$$

Proposition 2. *Consider a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$. The following conditions are equivalent:*

1. $LD(\mathcal{G}, o, \mathcal{D}) = 1$
2. *There exist $|\mathcal{D}|$ edge-independent paths in $\bar{\mathcal{G}}$ from o to $\bar{d}$*
3. *There exists a feasible o-$\bar{d}$ flow f on $\bar{\mathcal{G}}$ of value $\nu(f) = |\mathcal{D}|$ and such that $f_e \in \{0, 1\}$ for every $e \in \bar{\mathcal{E}}$.*

Moreover, given any flow f satisfying condition (3), the family of edges for which $f_e = 1$ are the support of $|\mathcal{D}|$ edge-independent paths in $\bar{\mathcal{G}}$ from o to $\bar{d}$ and, deleting the last edge in such paths from destinations d_i to $\bar{d}$, we obtain a 1-legible $(o, \mathcal{D})$-connecting set $\mathcal{P}$.

This result leads to an efficient way to establish whether $LD(\mathcal{G}, o, \mathcal{D}) = 1$ and construct a 1-legible $(o, \mathcal{D})$-connecting set $\mathcal{P}$. Using the well-known Ford-Fulkerson [19]'s algorithm, we can calculate a feasible o-$\bar{d}$ flow f^* of maximum value $\nu^* = \nu(f^*)$. By construction, we always have that $\nu^* \le |\mathcal{D}|$. It thus follows from Proposition 2 that $LD(\mathcal{G}, o, \mathcal{D}) = 1$ if and only if $\nu^* = |\mathcal{D}|$. When this is the case, the set $\mathcal{P}$ can then be easily constructed from the support of max flow f^*, i.e. by collecting the edges where the flow f^* runs.

Bernardini et al. [8] show that Problem 2 can also be approached by using network flow techniques, and in particular by solving a minimum-cost flow problem. Given the weight cost vector W on graph $\mathcal{G}$, let us extend it to a weight cost vector $\bar{W}$ on $\bar{\mathcal{G}}$ by giving zero cost to the edges connecting the destinations $\mathcal{D}$ to the fictitious destination $\bar{d}$ and maintaining the same cost to all the other edges. The *cost* of an o-$\bar{d}$ flow f on $\bar{\mathcal{G}}$ is defined as follows:

$$c(f) = \sum_{e \in \mathcal{E}} f(e) \bar{W}_e$$

The following result holds [8].

Proposition 3. *Given a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ with weight cost vector W, the minimum cost of a 1-legible $(o, \mathcal{D})$-connecting set of paths $\mathcal{P}$ in $\mathcal{G}$ is given by*

$$C^{min}_{(\mathcal{G},o,\mathcal{D})}(1) = \min_{f \;|\; \nu(f)=|\mathcal{D}|} c(f)$$

where the minimum on the right-hand side of the formula above is calculated over all feasible o-$\bar{d}$ flows with $\nu(f) = |\mathcal{D}|$, with the convention of being $+\infty$ if feasible flows do not exist.

Thanks to this result, the fast algorithms that have been developed to solve the minimum-cost flow problem can be applied to find solutions for Problem 2 when $k = 1$.

5.2. Reformulating s-legibility as 1-legibility

Bernardini et al. [8] show that checking s-legibility can be reduced to checking 1-legibility when we operate a suitable transformation of the original graph $\mathcal{G}$.

The key idea is the introduction of the so-called *s-legibility graph* $\mathcal{G}^{(s)}$, whose vertices consist of all walks of length $s-1$ in $\mathcal{G}$ and two vertices are connected if the corresponding walks overlap completely except for the first element of the first walk and last element of the second walk. While this will not suffice to solve the partial observability case, we briefly review this construction as some of the concepts and notation will constitute the building blocks of the more complex construction laid out in Section 7.

Given $\gamma = (e_1, \dots, e_s) \in \mathcal{Q}^{(s)}$, we put $\gamma_<$ for $(e_1, \dots, e_{s-1})$ and $\gamma_>$ for $(e_2, \dots, e_s)$. We then consider the following subsets of $\mathcal{Q}^{(s)}$, respectively, (i) walks that start in o but do not end in a destination; (ii) walks that end in a specific destination d_i but do not start in o; (iii) walks that start in o and end in a specific destination d_i; and (iv) walks that neither start in o nor end in $\mathcal{D}$:

$$
\begin{aligned}
&\text{(i)} \quad \mathcal{Q}^{(s)}_{(o} = \{(e_1,\dots,e_s) \in \mathcal{Q}^s \mid \theta(e_1) = o,\ \kappa(e_s) \notin \mathcal{D}\},\\
&\text{(ii)} \quad \mathcal{Q}^{(s)}_{d_i)} = \{(e_1,\dots,e_s) \in \mathcal{Q}^{(s)} \mid \theta(e_1) \neq o,\ \kappa(e_s) = d_i\},\\
&\text{(iii)} \quad \mathcal{Q}^{(s)}_{(o,d_i)} = \{(e_1,\dots,e_s) \in \mathcal{Q}^s \mid \theta(e_1) = o,\ \kappa(e_s) = d_i\},\\
&\text{(iv)} \quad \mathcal{Q}^{(s)}_{\bullet} = \mathcal{Q}^{(s)} \setminus \left(\mathcal{Q}^{(s)}_{(o} \bigcup_{d_i \in \mathcal{D}} \mathcal{Q}^{(s)}_{d_i)} \bigcup_{d_i \in \mathcal{D}} \mathcal{Q}^{(s)}_{o,d_i)} \right)
\end{aligned}
$$

We introduce a virtual origin $\bar{o}$ and a set of virtual destinations that is in one-to-one correspondence with the original set of destinations $\mathcal{D}$, i.e. $\bar{\mathcal{D}} = \{\bar{d}_1,\dots,\bar{d}_q\}$.

We are now ready to construct the s-Legibility directed (multi)graph $\mathcal{G}^{(s)} = (\mathcal{V}^{(s)}, \mathcal{E}^{(s)})$ where the sets $\mathcal{V}^{(s)}$ and $\mathcal{E}^{(s)}$ are as follows:

$$
\begin{aligned}
\mathcal{V}^{(s)} &= \mathcal{Q}^{(s-1)}_{\bullet} \cup \{\bar{o}\} \cup \bar{\mathcal{D}}\\
\mathcal{E}^{(s)} &= \mathcal{Q}^{(s)}
\end{aligned}
\tag{5}
$$

Tail and head functions are respectively $\theta : \mathcal{E}^{(s)} \to \mathcal{V}^{(s)}$ and $\kappa : \mathcal{E}^{(s)} \to \mathcal{V}^{(s)}$ and are defined based on whether $e \in \mathcal{E}^{(s)}$ is in $\mathcal{Q}^{(s)}_{\bullet}$, $\mathcal{Q}^{(s)}_{(o}$, $\mathcal{Q}^{(s)}_{D)}$ or $\mathcal{Q}^{(s)}_{(o,D)}$, as follows:

$$
\begin{aligned}
&\gamma \in \mathcal{Q}^{(s)}_{\bullet} \quad \theta(\gamma) = \gamma_<, \quad \kappa(\gamma) = \gamma_>\\
&\gamma \in \mathcal{Q}^{(s)}_{(o} \quad \theta(\gamma) = \bar{o}, \quad \kappa(\gamma) = \gamma_>\\
&\gamma \in \mathcal{Q}^{(s)}_{d_i)} \quad \theta(\gamma) = \gamma_<, \quad \kappa(\gamma) = \bar{d}_i\\
&\gamma \in \mathcal{Q}^{(s)}_{(o,d_i)} \quad \theta(\gamma) = \bar{o}, \quad \kappa(\gamma) = \bar{d}_i
\end{aligned}
$$

The node set $\mathcal{V}^{(s)}$ consists of (i) the set of subwalks of length $s-1$ in the original graph that do not start in o and do not end in any destination; (ii) the fictitious origin $\bar{o}$; and (iii) the fictitious set of destinations $\bar{\mathcal{D}}$. In the edge set $\mathcal{E}^{(s)}$, there are all walks γ of length s connecting pairs of subwalks of length $s-1$, i.e. $\gamma_<$ and $\gamma_>$, obtained from γ eliminating, respectively, the last and the first edge in γ. Notice that $\gamma_<$ and $\gamma_>$ overlap completely except for the first element in $\gamma_<$ and the last element in $\gamma_>$. There are a few exceptions: any walk γ that starts in o is considered as an edge starting from the virtual origin $\bar{o}$ and ending in $\gamma_>$; a walk that ends in a destination d_i is considered as an edge ending in the corresponding virtual destination $\bar{d}_i$ while starting in $\gamma_<$; and any walk γ that starts in o and ends in a destination d_i is considered as an edge starting from the virtual origin $\bar{o}$ and ending in the virtual destination $\bar{d}_i$.

There is a one-to-one mapping associating every walk γ in $\mathcal{G}$ from o to $\mathcal{D}$ with a walk $\omega = \Gamma(\gamma)$ in $\mathcal{G}^{(s)}$ from $\bar{o}$ to $\bar{\mathcal{D}}$:

$$\gamma = (e_1,\dots,e_l) \mapsto \omega = ((e_1,\dots,e_s),(e_2,\dots,e_{s+1}),\dots,)$$

The following theorem can be proven based on this mapping [8].

Theorem 1. *Given an LP instance $(\mathcal{G}, o, \mathcal{D})$ and a positive integer s, the following conditions are equivalent:*

1. *$(\mathcal{G}, o, \mathcal{D})$ is s-legible;*
2. *$(\mathcal{G}^{(s)}, \bar{o}, \bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{<s})$ is 1-legible.*

Based on Theorem 1, Problem 1 can be reformulated as checking 1-legibility of a sequence of legibility problems on $(\mathcal{G}^{(s)}, \bar{o}, \bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{<s})$) for $s = 1, 2, \dots$ until 1-legibility is found to hold. 1-legibility of each of these problems can be tackled as explained in Section 5.1. Bernardini et al. [8] also show how Problems 2 and 3 can be fruitfully addressed by leveraging these ideas.

Example 4 *(Continued).* Let us consider again graph $\mathcal{G}$ in Fig. 4 (repeated in Fig. 5 (a) for convenience) where all edges are observable and consider the LP instance $(\mathcal{G}, o, \mathcal{D})$. In Fig. 5 (b), we report the construction of $\mathcal{G}^{(2)}$. As the figure shows, the vertices of $\mathcal{G}^{(2)}$ correspond to the edges of $\mathcal{G}$, and the edges of $\mathcal{G}^{(2)}$ correspond to subwalks of length 2 in $\mathcal{G}$. We can find three walks to the three different destinations that do not overlap, for example,

$$\mathcal{P}' = \{((\alpha_1 x_1)(x_1\gamma)(\gamma x_3)(x_3\delta_1)), ((\alpha_2 x_2)(x_2\gamma)(\gamma x_4)(x_4\delta_2)), ((\alpha_3\beta_3)(\beta_3 x_5)(x_5\delta_3))\}$$

They are respectively depicted in violet, green, and orange in Fig. 5 (b). This tells us that the instance $(\mathcal{G}^{(2)}, o, \mathcal{D})$ is 1-legible and, consequently, the instance $(\mathcal{G}, o, \mathcal{D})$ is 2-legible. The following set of walks $\mathcal{P}$, obtained from $\mathcal{P}'$, exhibits 2-legibility:

$$\mathcal{P} = \{(\alpha_1 x_1 \gamma x_3 \delta_1), (\alpha_2 x_2 \gamma x_4 \delta_2), (\alpha_3 \beta_3 x_5 \delta_3)\}$$

They are correspondingly depicted in violet, green, and orange in Fig. 5 (a).

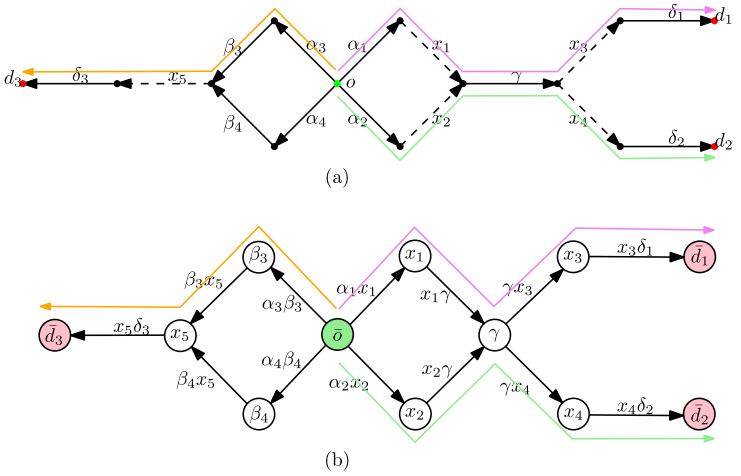


Fig. 5. (a) Graph $\mathcal{G}$; (b) Corresponding 2-Legibility graph $\mathcal{G}^{(2)}$. (Note that subwalks are indicated without parenthesis in the figures for brevity.)

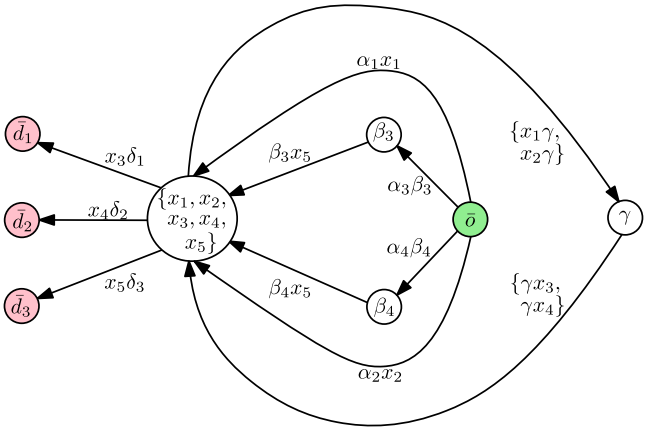


Fig. 6. Collapsed graph $\tilde{\mathcal{G}}^{(2)}$ relative to graph $\mathcal{G}$ in Fig. 5 (a). Note that we represent equivalence classes by using curly brackets.

6. Towards legibility under partial observability

Our goal in this section is to show that a simple generalization of the theory laid out in Section 5 does not work for tackling legibility under partial observability. We do that by offering several examples. A more sophisticated theory is needed for handling partial observability, which we present in Section 7.

Recall that our goal is to construct an auxiliary graph, the s-Legibility graph, so that, if this graph is 1-legible, the original graph is s-legible. In addition, we want that the $|\mathcal{D}|$ edge-disjoint walks that we find in the auxiliary graph can always be lifted to walks in the original graph that have a maximum overlap of $s-1$ edges. We need to avoid creating walks in the auxiliary graph that have no appropriate correspondent walks in the original graph.

As a first attempt to study legibility under partial observability, we propose a simple, intuitive modification of the s-Legibility graph approach used in the full observability case. It seems reasonable to represent two or more observably equivalent subwalks by using one vertex only since they are not distinguishable by the observer. We can think of them as being part of the same equivalence class. Hence, we build the s-Legibility directed (multi)graph $\mathcal{G}^{(s)} = (\mathcal{V}^{(s)}, \mathcal{E}^{(s)})$ as defined by Expression (5), and then we simply merge the vertices that correspond to observably equivalent subwalks by creating a vertex that represents their equivalence class.

Example *(Example 4 revisited).* Consider again the graph $\mathcal{G}$ represented in Fig. 4, where $\mathcal{D} = \{d_1, d_2, d_3\}$ and $\mathcal{E}_{nobs} = \{x_1, x_2, x_3, x_4, x_5\}$. Fig. 6 displays the graph $\tilde{\mathcal{G}}^{(2)}$ constructed by starting from the graph $\mathcal{G}^{(2)}$ (see Fig. 5 (b)) and then merging vertices and edges corresponding to observably equivalent subwalks of $\mathcal{G}$. Note that the LP instance $(\tilde{\mathcal{G}}^{(2)}, \bar{o}, \bar{\mathcal{D}})$ is 1-legible as the set of walks

$$\bar{\mathcal{P}} = \{((\alpha_1 x_1)(x_3\delta_1)), ((\alpha_2 x_2)(x_4\delta_2)), ((\alpha_3\beta_3)(\beta_3 x_5)(x_5\delta_3))\}$$

is $(\bar{o}, \bar{\mathcal{D}})$-connecting and 1-legible. If a theorem analogous to Theorem 1 was to hold in this case, we should obtain a set $\mathcal{P}$ of walks in $\mathcal{G}$ that is $(o, \mathcal{D})$-connecting and 2-legible by tracing back the walks in $\bar{\mathcal{P}}$ to walks in the original graph $\mathcal{G}$. Notice, however, that it is impossible to find a walk in $\mathcal{G}$ that corresponds to either the first or the second walk in $\bar{\mathcal{P}}$. Indeed, a walk in $\mathcal{G}$ corresponding

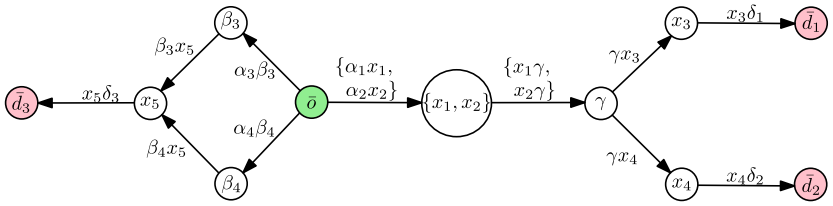


Fig. 7. Revised graph $\tilde{\mathcal{G}}^{(2)}$ relative to graph $\mathcal{G}$ in Fig. 5 (a).

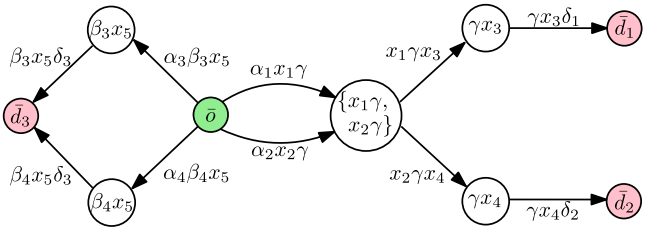


Fig. 8. Graph $\tilde{\mathcal{G}}^{(3)}$ relative to graph $\mathcal{G}$ in Fig. 5 (a).

to $((\alpha_1 x_1)(x_3\delta_1))$ would be $(\alpha_1 x_1 x_3\delta_1)$, but x_1 is not directly connected to x_3 in $\mathcal{G}$. The same also holds for the second walk, $((\alpha_2 x_2)(x_4\delta_2))$: a walk in $\mathcal{G}$ corresponding to it would be $(\alpha_2 x_2 x_4\delta_2)$, but x_2 is not directly connected to x_4 in $\mathcal{G}$. The problem here is that the compression of vertices in $\tilde{\mathcal{G}}^{(2)}$ has created additional, spurious walks, which do not correspond to any of the walks in $\mathcal{G}$. In particular, we have merged five unobservable edges, which lead to different vertices in the original graph, into a unique node.

To avoid the phenomenon seen in the example above, we need to reinforce our notion of equivalence. In particular, we consider an equivalence relation on subwalks that is stronger than the one in Definition 2 as, together with the condition of Definition 2, it also requires that the walks terminate in the same node.

Definition 7. Given $(e'_1,\dots,e'_s),(e''_1,\dots,e''_s)\in\mathcal{Q}^{(s)}$, we say that the two subwalks are *(strongly)* 0-*obs equivalent* and write

$$(e'_1,\dots,e'_s)\equiv_{0-obs}(e''_1,\dots,e''_s)$$

if they are observably equivalent and $\kappa(e'_s)=\kappa(e''_s)$.

The corresponding 0-obs (equivalence) class is denoted $[\gamma']_0=[\gamma'']_0$.

For example, in Fig. 5 (a), the two subwalks $(x_1\gamma)$ and $(x_2\gamma)$ are strongly 0-obs equivalent because they satisfy Definition 2 (they generate the same sequence of observations (γ)) and terminate in the same node. On the other end, the two subwalks (γx_3) and (γx_4) are not strongly 0-obs equivalent because, although they satisfy Definition 2 (again, they generate the same sequence of observations (γ)), they end in two different nodes.

We now apply this stronger notion of equivalence to the construction of the s-legibility graph for PO-LP instances by merging vertices that correspond to strongly observably equivalent subwalks into the same equivalence class. Let us see how this construction changes the s-legibility graph of our running example.

Example *(Example 4 revisited).* From graph $\mathcal{G}^{(2)}$ in Fig. 5 (b), we now obtain the graph $\tilde{\mathcal{G}}^{(2)}$ represented in Fig. 7. We only collapse vertices x_1 and x_2 this time because they are the only two observably equivalent subwalks that end in the same node. Thanks to our stronger notion of equivalence, walks in this new graph can always be lifted to walks in the original graph. The LP instance $(\tilde{\mathcal{G}}^{(2)},\bar{o},\bar{\mathcal{D}})$ is not 1-legible, which is in agreement with the PO-LP instance $(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})$ not being 2-legible.

However, this construction is problematic due to another issue. If we construct $\tilde{\mathcal{G}}^{(3)}$ by applying the same equivalence relation, we obtain the graph reported in Fig. 8. The LP instance $(\tilde{\mathcal{G}}^{(3)},\bar{o},\bar{\mathcal{D}})$ is 1-legible; see, for example, the following walks:

$$\bar{\mathcal{P}}=\{((\alpha_1 x_1\gamma)(x_1\gamma x_3)(\gamma x_3\delta_1)),((\alpha_2 x_2\gamma)(x_2\gamma x_4)(\gamma x_4\delta_2)),((\alpha_3\beta_3 x_5)(\beta_3 x_5\delta_3))\}$$

Following our theoretical framework, this would imply that the PO-LP instance has legibility 3, which is incorrect since the subwalks $(x_1\gamma x_3)$ and $(x_2\gamma x_4)$ are observably equivalent. The correct legibility of the PO-LP instance is 4, which is not revealed by the graph in Fig. 8.

The problem with the previous example is that the node $\{(x_1\gamma),(x_2\gamma)\}$ has two outgoing edges, $(x_1\gamma x_3)$ and $(x_1\gamma x_4)$, that are observably equivalent but end with two different unobservable edges, x_3 and x_4. The two edges $(x_1\gamma x_3)$ and $(x_1\gamma x_4)$ create two different walks to destinations $\bar{d}_1$ and $\bar{d}_2$, when in effect an observer following the actor would see the same γ both when the actor

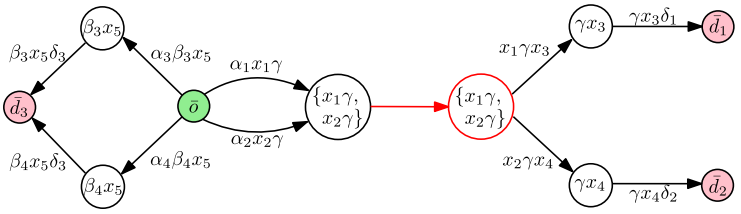


Fig. 9. Insertion of an auxiliary edge and node in correspondence with node $\{(x_1\gamma),(x_2\gamma)\}$ of the graph in Fig. 8, which has outgoing edges labeled with subwalks ($(x_1\gamma x_3)$ and $(x_2\gamma x_4)$) that end with unobservable edges (x_3 and x_4).

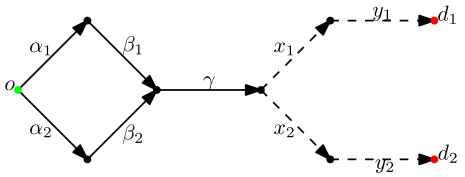


Fig. 10. Graph $\mathcal{G}$.

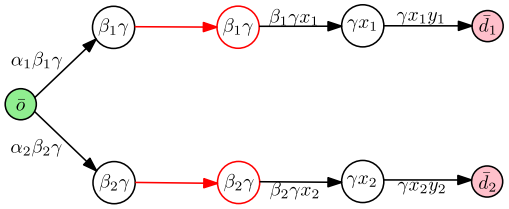


Fig. 11. Graph $\tilde{\mathcal{G}}^{(3)}$ relative to graph $\mathcal{G}$ in Fig. 10.

traverses $(x_1\gamma x_3)$ and when it traverses $(x_1\gamma x_4)$, and consequently it will not be able to tell where the actor aims to go. This type of problems can be overcome by inserting an *auxiliary edge* (and a corresponding auxiliary node) in the s-Legibility graph whenever a node such as $\{(x_1\gamma),(x_2\gamma)\}$ has outgoing edges labeled with subwalks that end with unobservable edges such as $(x_1\gamma x_3)$ and $(x_1\gamma x_4)$. The auxiliary edge creates a bottleneck in terms of legibility for the walks that end in $\bar{d}_1$ and $\bar{d}_2$ as it needs to be traversed to reach both destinations. The auxiliary edge increases the legibility delay by one by creating ambiguity in deciding whether an actor that passes through it aims to go to $\bar{d}_1$ or $\bar{d}_2$, reflecting the fact that observing γ only is not sufficient to decide the goal of the actor.

Example *(Example 4 revisited).* In Fig. 9, we show the insertion of an auxiliary edge and corresponding auxiliary node (represented in red) into the graph of Fig. 8. With this new edge, the instance is no longer 1-legible because the auxiliary edge needs to be traversed to reach both destinations $\bar{d}_1$ and $\bar{d}_2$. This solves the problem discussed in the graph in Fig. 8.

In the next section, we will prove that the insertion of auxiliary edges and corresponding nodes solves all issues arising when there are isolated unobservable edges in the original graph (in our case, x_1 and x_2). However, when there are adjacent unobservable edges, this artifice is no longer sufficient, as shown in the next example.

Example 5. Consider the graph $\mathcal{G}$ in Fig. 10 with $\mathcal{D} = \{d_1, d_2\}$ and $\mathcal{E}_{nobs} = \{x_1, x_2, y_1, y_2\}$. Suppose that it has been already established that $\mathcal{G}$ is not 1 or 2-legible, and we now want to determine 3-legibility. We construct $\tilde{\mathcal{G}}^{(3)}$. Following our previous observations, the graph is equipped with auxiliary nodes and edges as shown in Fig. 11. Note that the construction has created two completely independent walks towards the destinations $\bar{d}_1$ and $\bar{d}_2$. In other terms, the LP instance $(\tilde{\mathcal{G}}^{(3)}, \bar{o}, \bar{\mathcal{D}})$ results to be 1-legible. However, the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ is not 3-legible: the two subwalks $(\gamma x_1 y_1)$ and $(\gamma x_2 y_2)$ are observably equivalent and are the only subwalks of length 3 leading to the two destinations, respectively, d_1 and d_2.

The problem in the previous example emerges from the fact that the two subwalks $(\beta_1\gamma)$ and $(\beta_2\gamma)$, although not observably equivalent as β_1 and β_2 are distinct and observable, will eventually lead to observably equivalent subwalks if two unobservable edges are added to each of them sequentially ($(\gamma x_1 y_1)$ and $(\gamma x_2 y_2)$). To talk about these cases and avoid the related problem, we give the following definition that introduces a family of equivalence relations increasingly coarser than $\equiv_{0-obs}$. We use them to deal with sequences of adjacent unobservable edges.

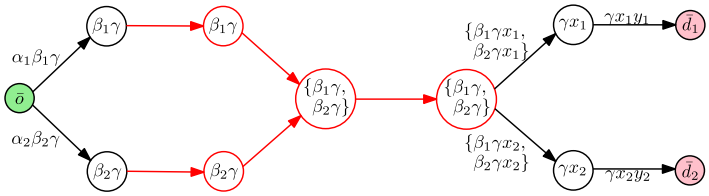


Fig. 12. Revised graph $\tilde{\mathcal{G}}^{(3)}$ relative to graph $\mathcal{G}$ in Fig. 10 obtained by introducing a node corresponding to the 2-obs class $[(\beta_1\gamma)]_2 = [(\beta_2\gamma)]_2$.

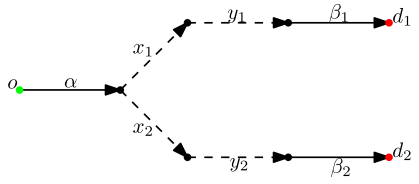


Fig. 13. Graph $\mathcal{G}$.

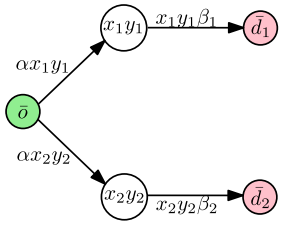


Fig. 14. Graph $\tilde{\mathcal{G}}^{(3)}$ relative to graph $\mathcal{G}$ in Fig. 13.

Definition 8. Given $k \geq 1$, we say that $\gamma' = (e'_1, \dots, e'_s)$ and $\gamma'' = (e''_1, \dots, e''_s)$ in $\mathcal{Q}^{(s)}$ are *k-obs equivalent* and write $\gamma' \equiv_{k-obs} \gamma''$ if

$$(e'_k, \dots, e'_s) \equiv_{0-obs} (e''_k, \dots, e''_s)$$

The corresponding k-obs (equivalence) class is denoted $[\gamma']_k = [\gamma'']_k$.

Two walks are k-obs equivalent if they are 0-obs equivalent starting from the index k. For example, in Fig. 10, the two subwalks $(\beta_1\gamma)$ and $(\beta_2\gamma)$ are 2-obs equivalent as they are 0-obs equivalent from index 2. Note that 0-obs and the 1-obs equivalence relations coincide. We keep both to simplify our theoretical construction in the next section.

Example *(Example 5 revisited).* We now consider not only 1-obs equivalent subwalks but also 2-obs equivalent subwalks (again, 1-obs equivalent subwalks are the same as 0-obs ones). In particular, we create a new node that corresponds to the 2-obs class $[(\beta_1\gamma)]_2 = [(\beta_2\gamma)]_2 = \{(\beta_1\gamma), (\beta_2\gamma)\}$, and we add it to $\tilde{\mathcal{G}}^{(3)}$ with incoming edges from both nodes $(\beta_1\gamma)$ and $(\beta_2\gamma)$. Now, we need to deal with the isolated unobservable edges x_1 and x_2. As explained above, we create an auxiliary edge and corresponding auxiliary node $\{(\beta_1\gamma), (\beta_2\gamma)\}$, similarly to what we did in Figs. 8 and 9. The construction leads to a new graph $\tilde{\mathcal{G}}^{(3)}$ illustrated in Fig. 12. Notice that now $\tilde{\mathcal{G}}^{(3)}$ is not 1-legible because the auxiliary edge creates a bottleneck.

Before illustrating the general construction in the next section, there is one more difficulty that concerns the origin, which we discuss in the next example.

Example 6. Consider graph $\mathcal{G}$ in Fig. 13 with $\mathcal{D} = \{d_1, d_2\}$ and $\mathcal{E}_{nobs} = \{x_1, x_2, y_1, y_2\}$. Suppose that it has been already established that $\mathcal{G}$ is not one or two legible, and we now want to determine 3-legibility. We construct $\tilde{\mathcal{G}}^{(3)}$ shown in Fig. 14. The LP instance $(\tilde{\mathcal{G}}^{(3)}, \bar{o}, \bar{\mathcal{D}})$ is 1-legible, but it is easy to see that the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ has legibility delay equal to 4 as the subwalks $(\alpha x_1 \beta_1)$ and $(\alpha x_2 \beta_2)$ do not allow us to disambiguate between $\bar{d}_1$ and $\bar{d}_2$. This indicates that the construction described so far leads to an incorrect result in this case.

We deal with this problem by treating subwalks starting in o and ending with an unobservable edge in a way different from our general construction. More specifically, we create an auxiliary node that corresponds to the subwalk of maximal length up to $s-1$ that starts in o and ends with an observable edge (α in Example 6) and a corresponding auxiliary edge from $\bar{o}$ to such an auxiliary node. We follow the regular construction explained above from that point on. Once again, the insertion of an auxiliary edge creates a bottleneck that impedes the multiplication of walks from the origin.

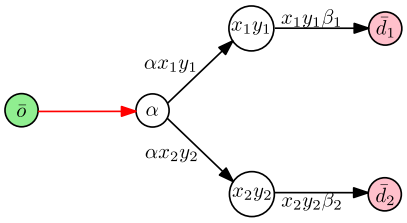


Fig. 15. Revised graph $\tilde{\mathcal{G}}^{(3)}$ relative to graph $\mathcal{G}$ in Fig. 13 after the insertion of an auxiliary edge.

Example *(Example 6 revisited).* The insertion of an auxiliary node and edge as explained above leads to the new graph $\tilde{\mathcal{G}}^{(3)}$ illustrated in Fig. 15. Notice that the LP instance $(\tilde{\mathcal{G}}^{(3)}, o, \mathcal{D})$ is no longer 1-legible.

In the next section, we systematize the artifices used in the examples above into a coherent theory to solve the problems enunciated in Section 4. Note that the main innovations to deal with partial observability are the introduction of equivalence classes at different levels of granularity to represent observably equivalent walks and the introduction of auxiliary edges and corresponding nodes to avoid the proliferation of walks in the s-Legibility graph that cannot be lifted to walks in the original graph.

7. Legibility under partial observability: theoretical construction

We are now ready to present formally the general construction of an s-Legibility graph under partial observability. This graph will allow us to check the s-legibility under partial observability of an instance by checking the 1-legibility under full observability of a related instance.

We denote by $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$, for $k \geq 0$, the set of k-obs classes of subwalks in $\mathcal{Q}^{(s)}$. Despite the fact that the 0-obs and the 1-obs equivalence relations coincide, we consider $\mathcal{Q}^{(s)}_{\equiv_{0-obs}}$ and $\mathcal{Q}^{(s)}_{\equiv_{1-obs}}$ as distinct sets. We have a sequence of projections, with the first one being a bijection:

$$\mathcal{Q}^{(s)}_{\equiv_{0-obs}} \to \mathcal{Q}^{(s)}_{\equiv_{1-obs}} \to \mathcal{Q}^{(s)}_{\equiv_{2-obs}} \to \cdots \tag{6}$$

given by

$$[\gamma]_0 \mapsto [\gamma]_1 \mapsto [\gamma]_2 \mapsto \cdots$$

If $\gamma = (e_1, \dots, e_s)$ and $k \geq 1$, based on Definitions 7 and 8, the k-obs class $[\gamma]_k$ is in one-to-one correspondence with the 1-obs class of the suffixes $[(e_k, \dots, e_s)]_1$. Following this, in our future derivations, we will identify a k-obs class $[\gamma]_k \in \mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ with a k'-obs class $[\gamma]_{k'} \in \mathcal{Q}^{(s')}_{\equiv_{k'-obs}}$ whenever $s' - k' = s - k$, and, given $\gamma = (e_1, \dots, e_s)$ and $\gamma' = (e'_1, \dots, e'_{s'})$, it holds that $[(e_k, \dots, e_s)]_1 = [(e'_{k'}, \dots, e'_{s'})]_1$.

Coherently with our notation, the following sets

$$\mathcal{Q}^{(s)}_{(o\equiv_{k-obs}}, \quad \mathcal{Q}^{(s)}_{d)\equiv_{k-obs}}, \quad \mathcal{Q}^{(s)}_{(o,d)\equiv_{k-obs}}, \quad \mathcal{Q}^{(s)}_{\bullet\equiv_{k-obs}}$$

indicate the subwalks in $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that, respectively, start in o but do not end in a destination, do not start in o and end in a destination d, start in o and end in d, and all the others (i.e. those that do not start in o and do not end in a destination). Finally, we let $\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$ and $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$ be the sets of those k-obs classes whose representative subwalks end with, respectively, an observable edge or an unobservable edge. We have $\mathcal{Q}^{(s)}_{\equiv_{k-obs}} = \mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}} \cup \mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$. Given $[\gamma]_k \in \mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$, we indicate with $[\gamma]^{\dashv}_k$ the class of maximal prefixes of $[\gamma]_k$ that end with an observable edge.

We are now ready to define the directed graph that allows us to transform s-legibility under partial observability into standard 1-legibility. We call this graph *s-Legibility graph under partial observability* and put

$$\mathcal{G}^{(s)}_{po} = (\mathcal{V}^{(s)}, \mathcal{E}^{(s)}_{bas} \cup \mathcal{E}^{(s)}_{aux})$$

where $\mathcal{E}^{(s)}_{bas}$ are *basic* edges derived from subwalks in $\mathcal{G}$ (hence, encoding walks in the original graph $\mathcal{G}$), while $\mathcal{E}^{(s)}_{aux}$ are extra *auxiliary* edges, which are needed to cope with partial observability. We first describe the node set $\mathcal{V}^{(s)}$.

$$\begin{aligned}\mathcal{V}^{(s)} \;=\; & \mathcal{Q}^{(s-1)}_{(o\equiv_{0-obs}} \cup \mathcal{Q}^{(s-1)}_{\bullet\equiv_{0-obs}} \cup \{\bar{o}\} \cup \bar{\mathcal{D}} \\ & \cup \bigcup_{r\leq s-1} \bigcup_{k=1}^{r} \left[\mathcal{Q}^{(r)\dashv}_{(o\equiv_{k-obs}} \bigcup_{d\in\mathcal{D}} \mathcal{Q}^{(r)\dashv}_{(o,d)\equiv_{1-obs}} \right] \cup \bigcup_{k=1}^{s-1} \mathcal{Q}^{(s-1)}_{\bullet\equiv_{k-obs}} \cup \\ & \cup \bigcup_{r\leq s-1} \bigcup_{k=1}^{r} \left[(\mathcal{Q}^{(r)\dashv}_{(o\equiv_{k-obs}})' \bigcup_{d\in\mathcal{D}} (\mathcal{Q}^{(r)\dashv}_{(o,d)\equiv_{1-obs}})' \right] \cup \bigcup_{k=1}^{s-1} (\mathcal{Q}^{(s-1)}_{\bullet\equiv_{k-obs}})'\end{aligned} \tag{7}$$

where, given a set A, we indicate with A' a distinct copy of A equipped with a canonical bijection $a \in A \mapsto a' \in A'$.

We now describe the set of basic edges:

$$\mathcal{E}^{(s)}_{bas} = \mathcal{Q}^{(s)\dashv}_{\equiv_{0-obs}} \cup \bigcup_{k=1}^{s-1} \mathcal{Q}^{(s)\nexists}_{\equiv_{k-obs}} \tag{8}$$

The edges insist on nodes as described below (with $\bar{d} \in \bar{D}$).

$$\begin{aligned}
&[\gamma]_0 \in \mathcal{Q}^{(s)\dashv}_{\bullet\equiv_{0-obs}} && \theta([\gamma]_0) = [\gamma_<]_0 && \kappa([\gamma]_0) = [\gamma_>]_0\\
&[\gamma]_0 \in \mathcal{Q}^{(s)\dashv}_{(o\equiv_{0-obs}} && \theta([\gamma]_0) = \bar{o} && \kappa([\gamma]_0) = [\gamma_>]_0\\
&[\gamma]_0 \in \mathcal{Q}^{(s)\dashv}_{d)\equiv_{0-obs}} && \theta([\gamma]_0) = [\gamma_<]_0 && \kappa([\gamma]_0) = \bar{d}\\
&[\gamma]_0 \in \mathcal{Q}^{(s)\dashv}_{(o,d)\equiv_{0-obs}} && \theta([\gamma]_0) = \bar{o} && \kappa([\gamma]_0) = \bar{d}\\
&[\gamma]_k \in \mathcal{Q}^{(s)\nexists}_{\bullet\equiv_{k-obs}} && \theta([\gamma]_k) = [\gamma_<]_k && \kappa([\gamma]_k) = [\gamma_>]_{k-1}\\
&[\gamma]_k \in \mathcal{Q}^{(s)\nexists}_{(o\equiv_{k-obs}} && \theta([\gamma]_k) = [\gamma^{\dashv}_<]_k && \kappa([\gamma]_k) = [\gamma_>]_{k-1}\\
&[\gamma]_1 \in \mathcal{Q}^{(s)\nexists}_{d)\equiv_{1-obs}} && \theta([\gamma]_1) = [\gamma_<]_1 && \kappa([\gamma]_1) = \bar{d}\\
&[\gamma]_1 \in \mathcal{Q}^{(s)\nexists}_{(o,d)\equiv_{1-obs}} && \theta([\gamma]_1) = [\gamma^{\dashv}_<]_1 && \kappa([\gamma]_1) = \bar{d}
\end{aligned} \tag{9}$$

Auxiliary edges are compactly described below as subgraphs of $\mathcal{G}^{(s)}_{po}$. Edges are represented by arrows.

- For every $[\gamma]_1 \in \mathcal{Q}^{(r)\dashv}_{(o\equiv_{1-obs}} \bigcup_{d\in D} \mathcal{Q}^{(r)\dashv}_{(o,d)\equiv_{1-obs}}$ with $r \le s-1$, we have:

$$\bar{o} \to [\gamma]'_1 \to [\gamma]_1 \cdots \to [\gamma]'_r \to [\gamma]_r \tag{10}$$

  The unique walks from $\bar{o}$ to $[\gamma]_k$, for $k \ge 1$, described by Expression (10) are denoted by $\lambda_k[\gamma]$.
- For every $[\gamma]_0 \in \mathcal{Q}^{(s-1)\dashv}_{\bullet\equiv_{0-obs}}$, we have:

$$[\gamma]_0 \to [\gamma]'_1 \to [\gamma]_1 \to \cdots \to [\gamma]'_{s-1} \to [\gamma]_{s-1} \tag{11}$$

  The unique walks from $[\gamma]_0$ to $[\gamma]_k$ described by Expression (11) are similarly denoted by $\lambda_k[\gamma]$.

If the original graph $\mathcal{G}$ is equipped with a weight vector $W \in \mathbb{R}^{\mathcal{E}}_+$, this is extended to a weight vector $W^{(s)}$ in the s-legibility graph $\mathcal{G}^{(s)}_{po}$ in the following way. We put a weight equal to 0 on every auxiliary edge, while weights on basic edges are attributed as shown below:

$$\begin{aligned}
&[\gamma]_k \in \mathcal{Q}^{(s)}_{\bullet\equiv_{k-obs}} \cup \mathcal{Q}^{(s)}_{d)\equiv_{k-obs}} && W^{(s)}_{[\gamma]_k} = \min_{\substack{(e_1,\dots,e_s)\in\mathcal{Q}^{(s)}\\ [(e_1,\dots,e_s)]_k = [\gamma]_k}} W_{e_s}\\
&[\gamma]_k \in \mathcal{Q}^{(s)}_{(o\equiv_{k-obs}} \cup \mathcal{Q}^{(s)}_{(o,d)\equiv_{k-obs}} && W^{(s)}_{[\gamma]_k} = \min_{\substack{(e_1,\dots,e_s)\in\mathcal{Q}^{(s)}\\ [(e_1,\dots,e_s)]_k = [\gamma]_k}} (W_{e_1} + \cdots W_{e_s})
\end{aligned} \tag{12}$$

From now on, we assume that the weights in $\mathcal{G}^{(s)}_{po}$ are defined as explained above. The corresponding weights for walks and costs for sets of walks are defined as in (3) and (4) and will be denoted, respectively, by the symbols $W^{(s)}$ and $C^{(s)}$.

Checking the s-legibility of the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, D)$ is equivalent to checking the 1-legibility of the LP instance $(\mathcal{G}^{(s)}_{po}, \bar{o}, \bar{D})$. This result is now proven by following several technical steps. We start with constructing walks in $\mathcal{G}^{(s)}_{po}$ from $\bar{o}$ to $\bar{D}$. We consider walks of the form

$$\omega = \lambda^0\omega_1\lambda^1\omega_2\cdots\lambda^{l-1}\omega_l \tag{13}$$

where each ω_t in ω is a walk in $\mathcal{G}^{(s)}_{po}$ composed of basic edges and each λ^t in ω is a walk in $\mathcal{G}^{(s)}_{po}$ composed of auxiliary edges only. It is convenient to be able to enumerate the basic edges composing the various ω_t in ω with a unique labeling γ^i, which we use in point (a) below.

(a) Each ω_t in ω presents the following structure:

$$\omega_t = [\gamma^{h_{t-1}+1}]_{k_t-1}[\gamma^{h_{t-1}+2}]_{k_t-2}\cdots[\gamma^{h_t}]_{\alpha_t} \tag{14}$$

where $1 \le k_t \le s$, $\alpha_t = 0$ if $t < l$, $\alpha_l \in \{0,1\}$. The indices h_t are recursively defined as follows:

$$h_0 = 0, \quad h_t = \begin{cases} h_{t-1} + k_t & \text{if } \alpha_t = 0\\ h_{t-1} + k_t - 1 & \text{if } \alpha_t = 1 \end{cases} \quad t \ge 1 \tag{15}$$

All edges that appear in ω_t are in $\mathcal{E}^{(s)}_{bas}$. Specifically, for all $t = 1,\dots,l$, and for all $j = 1,\dots,k_t$, we have that:

$$[\gamma^{h_{t-1}+j}]_{k_t-j} \in \begin{cases} \mathcal{Q}^{(s)\not\dashv}_{\equiv_{(k_t-j)-obs}} & \text{if } j < k_t \\ \mathcal{Q}^{(s)\dashv}_{\equiv_{0-obs}} & \text{if } j = k_t \end{cases}$$

Moreover, the first edge departs from the origin while the last one ends at a destination, namely,

$$[\gamma^1]_{k_1-1} \in \mathcal{Q}^{(s)}_{(o\equiv_{(k_1-1)-obs}}$$
$$[\gamma^{h_l}]_{\alpha_l} \in \mathcal{Q}^{(s)}_{d)\equiv_{\alpha_l-obs}} \; d \in \mathcal{D}$$

if $h_l > 1$. If, instead, $h_l = 1$ (there is just one basic edge), we have that

$$[\gamma^1]_{k_1-1} \in \mathcal{Q}^{(s)}_{(o,d)\equiv_{(k_1-1)-obs}} \; d \in \mathcal{D}$$

and $k_1 \in \{1,2\}$.
We further impose the following connecting rules among adjacent edges:

$$\begin{aligned} &[\gamma_>^{h_{t-1}+j}]_{k_t-j-1} = [\gamma_<^{h_{t-1}+j+1}]_{k_t-j-1} \; 1 \le j \le k_t - 1 \\ &[\gamma_>^{h_{t-1}}]_{k_t-1} = [\gamma_<^{h_{t-1}+1}]_{k_t-1} \end{aligned} \tag{16}$$

(b) The subwalks λ^t in ω are composed of auxiliary edges only as follows:

$$\lambda^0 = \begin{cases} \lambda_{k_1-1}[\gamma^0] & \text{if } k_1 > 1 \\ \emptyset & \text{if } k_1 = 1 \end{cases} \tag{17}$$

where $[\gamma^0]_1 \in \mathcal{Q}^{(r)\dashv}_{(o\equiv_{1-obs}}$ is such that

$$[\gamma_<^{1\dashv}]_{k_1-1} = [\gamma^0]_{k_1-1}$$

Moreover,

$$\lambda^t = \begin{cases} \lambda_{k_{t+1}-1}[\gamma_>^{h_t}] & \text{if } k_{t+1} > 1 \\ \emptyset & \text{if } k_{t+1} = 1 \end{cases} \tag{18}$$

Any instance of Expression (13) with all the objects and indices satisfying the above properties is a walk in $\mathcal{G}^{(s)}_{po}$, which can be proven by inspection and using the relations (9) and the definitions of the auxiliary walks in Expressions (10) and (11). The following proposition establishes that instances of Expression (13) are the only possible walks in $\mathcal{G}^{(s)}_{po}$ from $\bar{o}$ to the destinations in $\bar{\mathcal{D}}$.

Proposition 4. *Every walk in $\mathcal{G}^{(s)}_{po}$ from the origin $\bar{o}$ to a destination $\bar{d} \in \bar{\mathcal{D}}$ has the structure given in Expression (13). Its elements satisfy the properties in items (a) and (b) given above.*

Proof. Suppose we are given a walk ω in $\mathcal{G}^{(s)}_{po}$ from $\bar{o}$ to a $\bar{d} \in \bar{\mathcal{D}}$. We want to prove that ω presents the structure given in Expression (13). It follows from relations (9) and (10) that ω starts either with an edge $[\gamma^1]_0 \in \mathcal{Q}^{(s)\dashv}_{(o\equiv_{0-obs}}$ or with a walk $\lambda_k[\gamma^0]$ for some $[\gamma^0]_1 \in \mathcal{Q}^{(r)\dashv}_{(o\equiv_{1-obs}}$ and some $r \le s-1$. In the first case, we have $\lambda^0 = \emptyset$ and $\omega_1 = [\gamma^1]_0$. In the second case, we put $k_1 = k+1$ and $\lambda^0 = \lambda_k[\gamma^0]$. It follows from Expression (9) that, in this case, the next edge is a basic edge $[\gamma^1]_{k_1-1} \in \mathcal{Q}^{(s)\not\dashv}_{(o\equiv_{(k_1-1)-obs}}$ such that $[\gamma_<^{1\dashv}]_{k_1-1} = [\gamma^0]_{k_1-1}$. The terminating node $[\gamma_>^1]_{k_1-1} \in \mathcal{Q}^{(s-1)\not\dashv}_{\bullet\equiv_{(k_1-2)-obs}}$ will be necessarily followed by a sequence of edges $[\gamma^2]_{k_1-2}, \dots, [\gamma^{k_1}]_0$ such that $[\gamma_>^r]_{k_1-r-1} = [\gamma_<^{r+1}]_{k_1-r-1}$ for all $r = 2, \dots, k_1 - 1$. Notice that, in both cases, we end with an edge $[\gamma^{k_1}]_0$. If this edge corresponds to a subwalk terminating in a destination d, we have done; if not, we can follow a similar argument and deduce that the walk will necessarily continue with a λ^1, which is either an empty set or a sequence of auxiliary edges as in Expression (11) that starts in the node $[\gamma_>^{k_1}]_0$. □

7.1. Solving problems 1, 2 and 3 using the s-legibility graph

Our goal is now to establish a one-to-one correspondence between $(o, \mathcal{D})$-connecting set of walks in $\mathcal{G}$ and $(\bar{o}, \bar{\mathcal{D}})$-connecting set of walks in $\mathcal{G}^{(s)}_{po}$. The first step is to define a natural transformation between walks in $\mathcal{G}$ and $\mathcal{G}^{(s)}_{po}$.

Given an s-observable walk $\gamma = (e_1, \dots, e_l)$ in $\mathcal{G}$ from o to a destination d of length $l \ge s$, we construct a corresponding walk $\Gamma(\gamma)$ in $\mathcal{G}^{(s)}_{po}$, according to the structure in Expression (13) as follows. We let $s \le \zeta_1 < \zeta_2 < \cdots < \zeta_n$ be the positions greater than or equal to s corresponding to the observable edges, namely $e_{\zeta_k} \in \mathcal{E}_{obs}$ and $e_i \in \mathcal{E}_{nobs}$ for every $i > s$ with $i \ne \zeta_k$ for every k. We now define the following objects:

$$k_1 = \zeta_1 - s + 1, \quad k_t = \zeta_t - \zeta_{t-1}, \text{ for } t = 2, \dots, n$$
$$\gamma^r = (e_r, \dots, e_{s+r-1}), \text{ for } r = 1, \dots, n - s + 1$$
$$\gamma^0 = (e_1, \dots, e_{r_0}), \text{ where } r_0 = \max\{r \le s \ : \ e_r \in \mathcal{E}_{obs}\}$$

The fact that γ is s-observable implies that $1 \le k_t \le s$ and that γ^0 is well defined. If we define the indices h_t according to Expression (15) and the subwalks λ^t and ω^t according to Expressions (14), (17), and (18), we obtain that the compatible relations (16) are satisfied by construction and, consequently, the corresponding walk ω in Expression (13) is a walk in $\mathcal{G}_{po}^{(s)}$ connecting $\bar{o}$ to $\bar{d}$. We put $\Gamma(\gamma) = \omega$.

Proposition 5. *The map Γ has the following properties:*

1. *Given two s-observable walks $\gamma = (e_1, \dots, e_l)$, $\gamma' = (e'_1, \dots, e'_{l'})$ in $\mathcal{G}$, with $l, l' \ge s$, connecting o to destinations, respectively, d and d', we have that*
$$\Gamma(\gamma) = \Gamma(\gamma') \Rightarrow \begin{cases} l = l', \ d = d' \\ e_k \equiv_{0-obs} e'_k \ \forall k \ge s \end{cases}$$
2. *For any walk ω in $\mathcal{G}_{po}^{(s)}$ from $\bar{o}$ to some destination $\bar{d}$, there exists an s-observable walk γ in $\mathcal{G}$ from o to the corresponding destination d such that $\Gamma(\gamma) = \omega$.*
3. *For every s-observable walk γ, it holds that $W^{(s)}(\Gamma(\gamma)) \le W(\gamma)$. Moreover, the walk γ in item 2. can always be chosen in such a way that $W(\gamma) = W^{(s)}(\omega)$.*

Proof. 1. It follows from the definition of the map Γ that, if γ has length $l \ge s$, $\Gamma(\gamma)$ is composed of $l - s + 1$ basic edges. This implies that $l - s + 1 = l' - s + 1$, which yields $l = l'$. Moreover, if γ leads to d, then $\Gamma^f(\gamma)$ leads to $\bar{d}$. Hence, we have that $\bar{d} = \bar{d'}$, which entails that $d = d'$. Finally, the construction implies that $(e_r, \dots, e_{s+r-1}) \equiv_{k-obs} (e'_r, \dots, e'_{s+r-1})$ for some $k \le s - 1$ for every $r = 1, \dots, l - s + 1$. This, in particular, implies that $e_k \equiv_{0-obs} e'_k$ for every $k \ge s$.

2. By virtue of Proposition 4, ω is structured as in Expression (13). Let γ^r for $r = 1, \dots, h_l$ be subwalks in $\mathcal{Q}^{(s)}$ inducing the corresponding classes $[\gamma^r]_q$ used in the construction of ω. Represent $\gamma^r = (e^r_1, \dots, e^r_s)$ and finally consider the sequence of edges $\gamma = (e^1_1, \dots, e^1_s, e^2_s, \dots, e^{h_l}_s)$. We claim that γ is a walk in the original graph $\mathcal{G}$. Notice that $(e^1_1, \dots, e^1_s)$ is a walk in $\mathcal{G}$ by construction. Thanks to the concatenation constraints in Expression (16), we have that for every $r = 1, \dots, h_l - 1$, there exists $k \le s - 1$ such that $[\gamma^r_>]_k = [\gamma^{r+1}_<]_k$. By the definition of the k-obs equivalence relation, we have that $k(\gamma^r_>) = k(\gamma^{r+1}_<)$ and, consequently, $k(e^r_s) = k(e^{r+1}_{s-1})$. Since $k(e^{r+1}_{s-1}) = \theta(e^{r+1}_s)$ as γ^{r+1} is a feasible walk in $\mathcal{G}$, we also have that $k(e^r_s) = \theta(e^{r+1}_s)$ for every $r = 1, \dots, h_l - 1$. This says that γ is a walk in $\mathcal{G}$. It is s-observable because of the properties of ω: by construction, γ^1 possesses at least one observable edge and, in the structure of ω, there are at most $s - 1$ basic consecutive subwalks $[\gamma^r]_q \in \mathcal{E}^{(s)}_{bas}$ ending with unobservable edges. Finally, given Γ's definition, it follows that $\Gamma(\gamma) = \omega$.

3. The inequality $W^{(s)}(\Gamma(\gamma)) \le w(\gamma)$ is a direct consequence of the definition of weights in Expression (12) and the definition of Γ. Regarding the second point, this follows when, in the proof of item 2., we choose subwalks $\gamma^r = (e^r_1, \dots, e^r_s)$ that minimize the corresponding weights as expressed in relations (12). □

We are now ready to state the following conclusive results, which establish that checking s-legibility under partial observability of the graph $\mathcal{G}$ and checking 1-legibility under full observability of the s-Legibility graph $\mathcal{G}_{po}^{(s)}$ are equivalent.

Theorem 2. *The following facts hold true:*

1. *Consider a set $\mathcal{P}$ of walks in $\mathcal{G}$ from o to $\mathcal{D}$ (all of length not smaller than s) that are $(o, \mathcal{D})$-connecting and s-legible. Then, $\Gamma(\mathcal{P})$ is $(\bar{o}, \bar{\mathcal{D}})$-connecting and 1-legible. Moreover, $C^{(s)}(\Gamma(\mathcal{P})) \le C(\mathcal{P})$.*
2. *Consider a set $\mathcal{R}$ of walks in $\mathcal{G}_{po}^{(s)}$ from $\bar{o}$ to $\bar{\mathcal{D}}$ that are $(\bar{o}, \bar{\mathcal{D}})$-connecting and 1-legible. Then, any set $\mathcal{P}$ of s-observable walks in $\mathcal{G}$ from o to $\mathcal{D}$ such that $\Gamma(\mathcal{P}) = \mathcal{R}$ is $(o, \mathcal{D})$-connecting and s-legible. Moreover, there exists a set $\mathcal{P}$ such that $C^{(s)}(\Gamma(\mathcal{P})) = C(\mathcal{P})$.*

Proof. 1. The fact that $\mathcal{R} = \Gamma(\mathcal{P})$ is $(\bar{o}, \bar{\mathcal{D}})$-connecting follows from the definition of Γ. Suppose, by contradiction, that $\mathcal{R}$ is not 1-legible. This implies that there exist two walks $\gamma^1, \gamma^2 \in \mathcal{P}$ leading to two distinct destinations $d^1 \ne d^2$ such that the corresponding walks in $\mathcal{G}_{po}^{(s)}$ $\omega^1 = \Gamma(\gamma^1)$ and $\omega^2 = \Gamma(\gamma^2)$ share (at least) one edge. There are two possibilities: either the shared edge is auxiliary or basic. In the first case, given the structure of the preamble graphs (10) and (11), necessarily, there exist $(e^1_1, \dots, e^1_{r_1})$ and $(e^2_1, \dots, e^2_{r_2})$ that are subwalks of, respectively, γ^1 and γ^2 such that

$$[(e^1_1, \dots, e^1_{r_1})]_{k_1} = [(e^2_1, \dots, e^2_{r_2})]_{k_2}$$

where k_1 and k_2 are such that $r_1 - k_1 = r_2 - k_2$. In this way, we account for the possibility that one or both the subwalks are prefixes of length possibly smaller than $s - 1$. By construction, this means that these two subwalks are followed in both γ_1 and γ_2 by at least

$$r = k_1 + s - r_1 - 1 = k_2 + s - r_2 - 1$$

unobservable edges $f_1^1, \dots, f_r^1$ and $f_1^2, \dots, f_r^2$. The two subwalks of γ^1 and γ^2, respectively,

$$(e_{k_1}^1, \dots, e_{r_1}^1, f_1^1, \dots, f_r^1), \qquad (e_{k_2}^2, \dots, e_{r_2}^2, f_1^2, \dots, f_r^2)$$

have length equal to s and, by construction, are observably equivalent. This contradicts the assumption on $\mathcal{P}$. If, instead, the shared edge is basic, there exist $(e_1^1, \dots, e_s^1)$ and $(e_1^2, \dots, e_s^2)$ that are subwalks of, respectively, γ' and γ'' such that

$$[(e_1^1, \dots, e_s^1)]_k = [(e_1^2, \dots, e_s^2)]_k$$

for some $k \leq s-1$. By construction, this means that these two subwalks are followed, in γ' and γ'' respectively, by exactly $k-1$ unobservable edges. This leads, as in the previous case, to two subwalks of length s that are observably equivalent. This again contradicts the assumption on $\mathcal{P}$.

Finally, the cost inequality follows from the corresponding weight inequality in item 2. of Proposition 5.

2. Fix $\mathcal{P}$ as a set of s-observable walks in $\mathcal{G}$ from o to $\mathcal{D}$ such that $\Gamma(\mathcal{P}) = \mathcal{R}$. Based on Γ's definition, it follows that $\mathcal{P}$ is $(o, \mathcal{D})$-connecting. Suppose, by contradiction, that $\mathcal{P}$ is not observably s-legible and let $\gamma^1 = (e_1^1, \dots, e_{l_1}^1)$ and $\gamma^2 = (e_1^2, \dots, e_{l_2}^2) \in \mathcal{P}$ be two walks leading to two different destinations $d_1 \neq d_2$ such that, for some indices r_1, r_2, we have:

$$(e_{r_1}^1, \dots, e_{r_1+s-1}^1) =_{obs} (e_{r_2}^2, \dots, e_{r_2+s-1}^2) \tag{19}$$

Let $k \leq s-1$ be such that $e_{r_1+k}^1 = e_{r_2+k}^2 \in \mathcal{E}_{obs}$ is the last observable edge in the two subwalks. If $k = s-1$, i.e. this edge coincides with the last edge of the two subwalks, then

$$(e_{r_1}^1, \dots, e_{r_1+s-1}^1) \equiv_{0-obs} (e_{r_2}^2, \dots, e_{r_2+s-1}^2)$$

is a common edge in $\Gamma(\gamma_1)$ and $\Gamma(\gamma_2)$, contradicting the assumption of 1-legibility. If, instead, $k < s-1$, the two subwalks end with $s-1-k$ unobservable edges. Let us first assume $r_1 + k \geq s$ and $r_2 + k \geq s$. This implies the existence of basic edges in $\omega^i = \Gamma(\gamma^i)$ for $i = 1, 2$, respectively,

$$[(e_{\tilde{r}_1}^1, \dots, e_{r_1+k}^1)]_0, \qquad [(e_{\tilde{r}_2}^2, \dots, e_{r_2+k}^2)]_0$$

with $r_1 + k - \tilde{r}_1 = r_2 + k - \tilde{r}_2 = s-1$. According to the structure of walks in $\mathcal{G}_{po}^{(s)}$, these two basic edges are followed by auxiliary edges, respectively,

$$\begin{array}{ll}
[(e_{\tilde{r}_1+1}^1, \dots, e_{r_1+k}^1)]_0 & [(e_{\tilde{r}_2+1}^2, \dots, e_{r_2+k}^2)]_0 \\
\downarrow & \downarrow \\
[(e_{\tilde{r}_1+1}^1, \dots, e_{r_1+k}^1)]'_1 & [(e_{\tilde{r}_2+1}^2, \dots, e_{r_2+k}^2)]'_1 \\
\downarrow & \downarrow \\
\vdots & \vdots \\
\downarrow & \downarrow \\
[(e_{\tilde{r}_1+1}^1, \dots, e_{r_1+k}^1)]'_{h_1} & [(e_{\tilde{r}_2+1}^2, \dots, e_{r_2+k}^2)']_{h_2} \\
\downarrow & \downarrow \\
[(e_{\tilde{r}_1+1}^1, \dots, e_{r_1+k}^1)]_{h_1} & [(e_{\tilde{r}_2+1}^2, \dots, e_{r_2+k}^2)]_{h_2}
\end{array}$$

with $h_1, h_2 \geq s-1-k$. From Expression (19) and the fact that $e_{r_1+k}^1 = e_{r_2+k}^2 \in \mathcal{E}_{obs}$ is the last observable edge in the two subwalks, it follows that

$$[(e_{\tilde{r}_1+1}^1, \dots, e_{r_1+k}^1)]_{s-1-k} = [(e_{\tilde{r}_2+1}^2, \dots, e_{r_2+k}^2)]_{s-1-k}$$

which implies that there must be a common auxiliary edge in the two preamble structures.

This contradicts the assumption that $\mathcal{R}$ is 1-legible.

We finally need to consider the case when $r_i + k < s$ for one or both $i \in \{1, 2\}$. Assume this happens for $i = 1$ only. Then, the first basic edge in ω^1 is

$$[(e_1^1, \dots, e_s^1)]_{k_1}$$

where $k_1 \geq r_1$ is the number of consecutive unobservable edges in ω^1 starting with e_s^1. This basic edge is preceded, according to Expression (13), by a sequence of auxiliary edges as follows:

$$\bar{o} \to [(e_1^1, \dots, e_{r_1+k}^1)]'_1 \to \cdots \to [(e_1^1, \dots, e_{r_1+k}^1)]'_{k_1} \to [(e_1^1, \dots, e_{r_1+k}^1)]_{k_1}$$

Notice that

$$\begin{aligned}
&[(e_1^1, \dots, e_{r_1+k}^1)]_{r_1} = [(e_{r_1}^1, \dots, e_{r_1+k}^1)]_1 \\
&[(e_{\tilde{r}_2+1}^2, \dots, e_{r_2+k}^2)]_{s-1-k} = [(e_{r_2}^2, \dots, e_{r_2+k}^2)]_1
\end{aligned} \tag{20}$$

Relations (20) and (19) yield

$$[(e^1_1,\dots,e^1_{r_1+k})]_{r_1} = [(e^2_{r_2+1},\dots,e^2_{r_2+k})]_{s-1-k}$$

and thus again ω^1 and ω^2 would possess a common auxiliary edge. A similar argument holds for the other cases.

Finally, we can arrive at the cost equality between $\mathcal{P}$ and $\mathcal{R}$ if we make sure that $\mathcal{P}$ is constructed accordingly to item 2. in Proposition 5. Precisely, for every $\omega \in \mathcal{R}$, we choose a γ walk in $\mathcal{G}$ such that $\Gamma(\gamma) = \omega$ and $W(\gamma) = W^{(s)}(\gamma)$, and we set $\mathcal{P}$ to be the family of such walks γ. □

We are now ready to tackle Problems 1 to 3 in the general case. We show below how the graph constructions above allow us to solve these problems by essentially reducing them to the basic case of 1-legibility. We start with a direct consequence of Theorem 2, which tells us that checking s-legibility on the graph $\mathcal{G}$ can be transformed in checking 1-legibility on the new graph $\mathcal{G}^{(s)}_{po}$.

Corollary 1. *Given a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ and a positive integer s, the following two conditions are equivalent:*

1. *$(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ is s-legible;*
2. *$(\mathcal{G}^{(s)}_{po}, \bar{o}, \bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{<s})$ is 1-legible.*

Moreover, assume that the above conditions are satisfied, $\mathcal{P}_{<s}$ is any set of walks in $\mathcal{G}$ (all of length less than s) that is $(o, \mathcal{D}_{<s})$-connecting, and $\mathcal{R}$ is any 1-legible $(\bar{o}, \bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{<s})$-connecting set of walks in $\mathcal{G}^{(s)}_{po}$. Then, for every subset of walks $\mathcal{P}$ in $\mathcal{G}$ such that $\Gamma(\mathcal{P}) = \mathcal{R}$, we have that $\mathcal{P} \cup \mathcal{P}_{<s}$ is s-legible and $(o, \mathcal{D})$-connecting.

Proof. The statement is a direct consequence of Theorem 2 and of Remark 1. □

Problem 1 can be solved iteratively by checking s-legibility for increasing values of $s = 1, 2, \dots$ until we find that the property holds. For each s, thanks to Corollary 1 and Proposition 2, we reduce the legibility check to the computation of a max flow problem.

Problem 2 can also be addressed by making use of the auxiliary graphs $\mathcal{G}^{(s)}_{po}$. However, the presence of destinations that have a distance from the origin smaller than s has to be handled in a slightly different way. This is because it is not known a priori if the set that minimizes cost will contain walks of length smaller than s or longer walks (which might have a smaller cost). As walks shorter than s are not represented within $\mathcal{G}^{(s)}_{po}$, we need to incorporate them in it. Hence, we augment $\mathcal{G}^{(s)}_{po}$ by adding a family of extra edges. Precisely, for every destination d in $\mathcal{D}_{<s}$, we add one extra edge e_d that connects $\bar{o}$ to $\bar{d}$ such that $\theta(e_d) = \bar{o}$ and $k(e_d) = \bar{d}$, and we equip it with a weight $W^{(s)}_{e_d}$ defined as follows:

$$W^{(s)}_{e_d} = \min\{W(\gamma) \mid \gamma \text{ walk in } \mathcal{G} \text{ from } o \text{ to } d,\ l_\gamma < s\} \tag{21}$$

We also choose, for every $d \in \mathcal{D}_{<s}$, a walk

$$\gamma^d \in \operatorname{argmin}\{W(\gamma) \mid \gamma \text{ walk in } \mathcal{G} \text{ from } o \text{ to } d,\ l_\gamma < s\}$$

We indicate with the symbol $\bar{\mathcal{G}}^{(s)}_{po}$ this augmented version of $\mathcal{G}^{(s)}_{po}$.

Given a $(\bar{o}, \bar{\mathcal{D}})$-connecting set of walks $\mathcal{R}$ in $\bar{\mathcal{G}}^{(s)}_{po}$, we associate an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}_{\mathcal{R}}$ as follows. We define $\mathcal{D}' = \{d \in \mathcal{D}_{<s} \mid e_d \in \mathcal{R}\}$ and put

$$\mathcal{R}' = \{e_d \mid d \in \mathcal{D}'\} \quad \mathcal{R}'' = \mathcal{R} \setminus \mathcal{R}'$$

We now take $\mathcal{P}' = \{\gamma^d \mid d \in \mathcal{D}'\}$ and $\mathcal{P}''$ as any set of s-observable walks in $\mathcal{G}$ from o to $\mathcal{D}$ such that $\Gamma(\mathcal{P}'') = \mathcal{R}''$. Finally, we define

$$\mathcal{P}_{\mathcal{R}} = \mathcal{P}' \cup \mathcal{P}''$$

and notice that $\mathcal{P}$ is $(o, \mathcal{D})$-connecting and s-legible by construction and by item 2. of Theorem 2. The following result holds.

Corollary 2. *Consider a PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ and, given $k > 1$, the auxiliary legibility problem instance $(\bar{\mathcal{G}}^{(k)}, \bar{o}, \bar{\mathcal{D}})$. Then,*

$$C^{min}_{(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})}(k) = C^{min}_{(\bar{\mathcal{G}}^{(k)}_{po}, \bar{o}, \bar{\mathcal{D}})}(1) \tag{22}$$

Moreover, if $\mathcal{R}$ is any 1-legible $(\bar{o}, \bar{\mathcal{D}})$-connecting set of walks that minimizes the cost $C^{(s)}$, then $\mathcal{P}_{\mathcal{R}}$ is a s-legible $(\bar{o}, \bar{\mathcal{D}})$-connecting set of walks that minimizes the cost C.

Problem 3 can be solved by exploiting its duality with Problem 2 as explained in Bernardini et al. [8]. Given a budget B, the goal is to compute $LD^{min}_{(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})}(B)$. We can proceed as follows. We start from the legibility delay $s_1 = LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$ and compute, using the approach above, $B_1 = C^{min}_{(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})}(s_1)$. We put

$$LD^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})}(B) = s_1 \quad \forall B \geq B_1$$

We then take the next $s_2 > s_1$ such that $B_2 = C^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})}(s_2) < B_1$ and put

$$LD^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})}(B) = s_2 \quad \forall B_2 \leq B < B_1$$

We iterate in this way until the budget reaches the minimum value that is necessary to obtain an $(o,\mathcal{D})$-connecting set of walks. We obtain a step function for $LD^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})}(B)$, where each cost interval corresponds to the smallest legibility delay s that can be found by incurring that cost.

8. Algorithms

Given a *PO-LP* instance, we use an iterative approach for solving Problems 1, 2, and 3. At each step $s = 1,\dots,k$, we construct the graph $\mathcal{G}^{(s)}_{po}$ and consider the instance of the legibility problem under partial observability $(\mathcal{G}^{(s)}_{po}, \bar{o}, \bar{\mathcal{D}} \setminus \bar{\mathcal{D}}_{<s})$. We convert the graph $\mathcal{G}^{(s)}_{po}$ into a flow network $\mathcal{N}^{(s)}$ and solve a maximum flow problem (for solving Problem 1) or a minimum cost problem (for solving Problems 2 and 3). With respect to the original technique presented by Bernardini et al. [8], the changes revolve around the challenges of correctly encoding a partially observable environment into a s-legibility graph $\mathcal{G}^{(s)}_{po}$. Hence, in this section, we focus on describing the construction of the graph $\mathcal{G}^{(s)}_{po}$ for PO-LP instances $(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})$. We also analyze the complexity of this construction.

The case of $s = 1$ is simpler than the others. Recall that, as mentioned in Section 4, the concept of 1-legibility for a PO-LP instance $(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})$ is equivalent to the concept of 1-legibility for the corresponding LP instance $(\mathcal{G},o,\mathcal{D})$. We construct the graph $\mathcal{G}^{(1)}_{po}$ by simply removing all unobservable edges from $\mathcal{G}$ and adding edges from each destination in $\mathcal{D}$ to $\bar{d}$. We then use the resulting graph to construct the flow network and run the appropriate algorithms to solve Problems 1, 2 and 3 in this case.

To construct the graph $\mathcal{G}^{(s)}_{po}$ for $s > 1$, we follow Expressions (7), (9), (10) and (11) as presented in Section 7. To do so, we need to obtain the set of equivalence classes that define the set of nodes $\mathcal{V}^{(s)}$ and the set of edges $\mathcal{E}^{(s)} = \mathcal{E}^{(s)}_{bas} \cup \mathcal{E}^{(s)}_{aux}$. Constructing the graph $\mathcal{G}^{(s)}_{po}$ becomes trivial once we obtain all the sets of k-observable classes of subwalks of length s with k ranging from 0 to $s-1$, $\bigcup_{k=0}^{s-1} \mathcal{Q}^{(s)}_{\equiv_{k-obs}}$.

The generation of a set $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ entails comparing each subwalk $\gamma \in \mathcal{Q}^{(s)}$ with other subwalks in $\mathcal{Q}^{(s)}$, determining whether they are (strongly) k-observably equivalent, and then storing γ in the appropriate equivalence class. Note that, to build the graph $\mathcal{G}^{(s)}_{po}$, we might need to use most of the equivalent classes for k ranging from 0 to $s-1$, depending on the number of consecutive unobservable edges following a given subwalk (see Expressions (9), (10) and (11)). Since checking all these equivalences can become computationally taxing, we present Algorithms 1 and 2, which deal with this operation efficiently by avoiding redundant checks between subwalks whose equivalence can be directly inferred from previous checks.

For two subwalks to be strongly observably equivalent, they must terminate at the same vertex (see Definition 7). Thus, optimizing the search for equivalence classes can be achieved by comparing subwalks only if this condition holds. Another optimization method is based on exploiting the equivalence property of the subwalks from the same k-observable class. Suppose we want to check if a subwalk γ is part of an equivalence class along with other subwalks. We check if there exists any k-observable class $[\gamma']_k$ ending in the same node as γ. If there is, we only need to check γ against one of the members $\gamma' \in [\gamma']_k$. If we cannot find any existing class that is k-obs equivalent with γ, we create the single-element k-observable class $[\gamma]_k$.

When generating $\mathcal{G}^{(s)}_{po}$, we can encounter nodes representing subwalks with length smaller than $s-1$ (see Expression (7)). This requires us to verify the equivalence between subwalks of different lengths. Since the subwalks with length smaller than $s-1$ occur only in the immediate vicinity of $\bar{o}$, we check their equivalence against the subwalks of length $s-1$ after generating the children of $\bar{o}$ and their associated auxiliary nodes, following the same principles described above.

Before starting the iterative process, we define the base set $\mathcal{Q}^{(1)}_{\equiv_{1-obs}}$ from which $\bigcup_{k=0}^{2} \mathcal{Q}^{(2)}_{\equiv_{k-obs}}$ is derived:

$$\begin{aligned}
\mathcal{Q}^{(1)\dashv}_{\equiv_{0-obs}} &= \mathcal{Q}^{(1)\dashv}_{\equiv_{1-obs}} = \{e \mid e \in \mathcal{E}_{obs}\} \\
\mathcal{Q}^{(1)\not\dashv}_{\equiv_{0-obs}} &= \mathcal{Q}^{(1)\not\dashv}_{\equiv_{1-obs}} = \{e \mid e \in \mathcal{E}_{nobs}\} \\
\mathcal{Q}^{(1)}_{\equiv_{0-obs}} &= \mathcal{Q}^{(1)}_{\equiv_{1-obs}} = \mathcal{Q}^{(1)\dashv}_{\equiv_{1-obs}} \cup \mathcal{Q}^{(1)\not\dashv}_{\equiv_{1-obs}}
\end{aligned}$$

Function $Construct\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ in Algorithm 1 generates $\bigcup_{k=0}^{s} \mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ based on the set $\mathcal{Q}^{(s-1)}$ obtained at step $s-1$. Note that $\mathcal{Q}^{(s-1)}$ is formed by all individual subwalks from a set $\mathcal{Q}^{(s-1)}_{\equiv_{k-obs}}$. We also calculate the s-observable classes of subwalks of length s, since this information is necessary to construct the $\mathcal{G}^{(s+1)}_{po}$ (see as an example Expression (7)). The function takes as input the set of observable edges, the legibility delay s, and the set $\mathcal{Q}^{(s-1)}$.

Algorithm 1 is composed of four nested loops. At line 1, it traverses the vertices $v \in \mathcal{V}$, and at line 2, it iterates through all subwalks $\gamma \in \mathcal{Q}^{(s-1)}$ that terminate in a given vertex v. Then, at line 3, it searches for each outgoing edge of v to create (at line 4) a new subwalk γ' of length s. In the innermost loop, at line 5, function $Check\gamma_k Equiv$ (Algorithm 2) verifies the k-observable class membership of each γ'. At lines 6-9, the set $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ is initialized if it has not yet been created, together with its component sets of

Algorithm 1: $Construct\mathcal{Q}^{(s)}_{\equiv_{k-obs}}(\mathcal{V}, \mathcal{E}_{obs}, s, \mathcal{Q}^{(s-1)})$.

```
1  foreach v ∈ V do
2      foreach γ ∈ Q^(s-1) | κ(γ) = v do
3          foreach e_s ∈ out(v) do
4              γ' ← Concatenate(γ, e_s)
5              for k ← 1 to s do
6                  if ∄ Q^(s)_{≡_{k-obs}} then
7                      Q^(s)⊣_{≡_{k-obs}} ← ∅
8                      Q^(s)↯_{≡_{k-obs}} ← ∅
9                      Q^(s)_{≡_{k-obs}} ← ∅
10                 if e_s ∈ E_obs then
11                     Q^(s)⊣_{≡_{k-obs}} ← Check γ_k Equiv(k, γ', Q^(s)⊣_{≡_{k-obs}})
12                 else
13                     Q^(s)↯_{≡_{k-obs}} ← Check γ_k Equiv(k, γ', Q^(s)↯_{≡_{k-obs}})
14 for k ← 1 to s do
15     Q^(s)_{≡_{k-obs}} ← Q^(s)⊣_{≡_{k-obs}} ∪ Q^(s)↯_{≡_{k-obs}}
16 Q^(s)⊣_{≡_{0-obs}} ← Q^(s)⊣_{≡_{1-obs}}
17 Q^(s)↯_{≡_{0-obs}} ← Q^(s)↯_{≡_{1-obs}}
18 Q^(s)_{≡_{0-obs}} ← Q^(s)⊣_{≡_{0-obs}} ∪ Q^(s)↯_{≡_{0-obs}}
19 return ⋃_{k=0}^{s} Q^(s)_{≡_{k-obs}}
```

k-observable classes whose representative subwalks end with an observable edge or an unobservable edge, i.e. $\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$ and $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$, respectively. The algorithm checks whether γ' ends in an observable edge or not, and it passes to $Check\gamma_k Equiv$ the appropriate set of equivalence classes, at lines 10-13. We distinctly populate the sets $\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$ and $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$ since the rules for constructing $\mathcal{G}^{(s)}_{po}$ are different depending on the observability of the last edge in a subwalk. After updating all sets $\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$ and $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$, we iterate from 1 to s and assign to each $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ the union of its component sets, at lines 14-15. Given that $\mathcal{Q}^{(s)}_{\equiv_{0-obs}}$ and $\mathcal{Q}^{(s)}_{\equiv_{1-obs}}$ coincide, it is sufficient to generate $\mathcal{Q}^{(s)\dashv}_{\equiv_{1-obs}}$ and $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{1-obs}}$ inside the nested loops and, then, at lines 16-17, we assign them to their 0-obs homologous classes. Their union is then assigned to the set $\mathcal{Q}^{(s)}_{\equiv_{0-obs}}$, at line 18.

Algorithm 2: $Check\gamma_k Equiv(k, \gamma, \mathcal{Q}^{(s)-}_{\equiv_{k-obs}})$.

```
1  equiv_found ← ⊥
2  foreach [γ']_k ∈ Q^(s)-_{≡_{k-obs}} do
3      γ' ←R [γ']_k  // any γ' in [γ']_k
4      if κ(γ) = κ(γ') and γ ≡_{k-obs} γ' then
5          [γ']_k ← [γ']_k ∪ {γ}
6          equiv_found ← ⊤
7          break
8  if ¬equiv_found then
9      [γ]_k ← {γ}
10     Q^(s)-_{≡_{k-obs}} ← Q^(s)-_{≡_{k-obs}} ∪ [γ]_k
11 return Q^(s)-_{≡_{k-obs}}
```

Algorithm 2 is responsible for populating all sets $\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$ and $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$. The function $Check\gamma_k Equiv$ takes three inputs: a subwalk suffix index k, a subwalk γ and a set $\mathcal{Q}^{(s)-}_{\equiv_{k-obs}}$ to be updated. We use the variable $\mathcal{Q}^{(s)-}_{\equiv_{k-obs}}$ as a holding variable for either set $\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$ or $\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$. The variable *equiv_found* determines when an equivalence is found, and at line 1, it is initially set to false. The algorithm proceeds by iterating through the equivalence classes of type $[\gamma']_k$ from $\mathcal{Q}^{(s)-}_{\equiv_{k-obs}}$, as shown in line 2. Then, at line 3, any subwalk that is part of a given equivalence class $[\gamma']_k$ is assigned to γ'. Note that here $[\gamma']_k$ represents all subwalks γ'' that form a k-observable equivalence relation with a representative subwalk γ'. At line 4, the algorithm checks if both γ and γ' terminate in the same vertex and if they are k-obs equivalent. If these conditions hold, $[\gamma']_k$ stores the new-found k-observable subwalk γ and the for loop breaks. The variable *equiv_found* is set to true to indicate that an equivalence has been found. If no equivalence is found, then, the equivalence class $[\gamma]_k$ is created and added to $\mathcal{Q}^{(s)-}_{\equiv_{k-obs}}$, at lines 8-10.

During the construction of $\mathcal{G}^{(s)}_{po}$, when generating auxiliary nodes, we will only use the k-observable classes up to the level corresponding to the number of unobservable consecutive edges following the given subwalk. Hence, not all subwalks in $\bigcup_{k=0}^{s} \mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ will be used in $\mathcal{G}^{(s)}_{po}$.

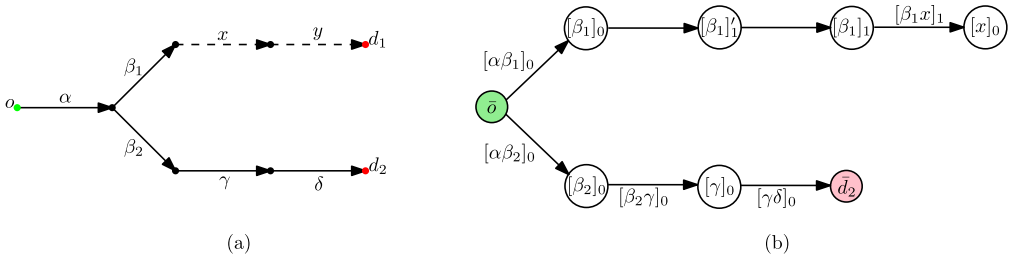


Fig. 16. Example of a graph $\mathcal{G}$ where not all destinations are reachable in the associated $\mathcal{G}^{(2)}_{po}$: Figure (a) represents the original graph $\mathcal{G}$ and Figure (b) is the corresponding $\mathcal{G}^{(2)}_{po}$.

One final optimization involves avoiding checking for s-legibility altogether when all possible walks towards a destination $d \in \mathcal{D}$ contain one or more subwalks of consecutive unobservable edges of length greater than $s-1$ (see the walk terminating in d_1 in Fig. 16(a)). In this situation, there is no $(o,\mathcal{D})$-connecting set of walks $\mathcal{P}$ that would satisfy condition (i) of Definition 4, i.e. any walk γ leading to d cannot be s-observable. An efficient way to verify this is to check that each node $\bar{d} \in \bar{\mathcal{D}}$ exists in $\mathcal{G}^{(s)}_{po}$ right after constructing $\mathcal{G}^{(s)}_{po}$ and before defining it as a flow network problem.

Example 7. Let us consider the graph $\mathcal{G}$ depicted in Fig. 16(a), where $\mathcal{D} = \{d_1, d_2\}$ and $\mathcal{E}_{nobs} = \{x, y\}$. Notice that there is a single possible $(o,\mathcal{D})$-connecting set of walks $\mathcal{P} = \{(\alpha\beta_1 xy), (\alpha\beta_2\gamma\delta)\}$.

It is easy to see that $\mathcal{G}$ cannot be 1-legible nor 2-legible, since α is a common edge and the walk leading to d_1 is not 2-observable due to the subwalk (xy). Nonetheless, let us consider $\mathcal{G}^{(2)}_{po}$ displayed in Fig. 16(b) to understand better how the algorithm avoids calculating the optimal flow in this type of case.

The walk $(\alpha\beta_2\gamma\delta)$ is fully observable, hence it is encoded in subwalks from 0-obs equivalence classes. Since the walk $(\alpha\beta_1 xy)$ contains a subwalk of length 2 of only unobservable edges, we need to introduce the auxiliary constructions as described in Expression (11), which are then continued by applying the fifth rule from Expression (9). We see that, by construction, $\mathcal{G}^{(2)}_{po}$ cannot contain $\bar{d}_1$. We can then infer that an optimal flow equal to 2 cannot be found in the corresponding flow network of $\mathcal{G}^{(2)}_{po}$.

This example demonstrates as well how the auxiliary constructions correctly capture that the walk to d_1 is not 2-observable, therefore, $\mathcal{P}$ cannot be 2-legible.

Note that, if a walk towards a destination d is shorter than the legibility delay, we still represent that walk in $\mathcal{G}^{(s)}_{po}$ and connect it to $\bar{d}$. Hence, this situation does not affect our proposed verification. In the next section, we present the effect of implementing this extra pre-analysis of $\mathcal{G}^{(s)}_{po}$ on the time complexity.

8.1. Complexity

Here, we discuss the worst-case complexity of the proposed technique to construct $\mathcal{G}^{(s)}_{po}$, assuming a uniform graph $\mathcal{G}$ of branching factor B for simplicity. The complexity is dominated by the four nested loops in Algorithm 1 and the number of iterations inside the function $Check\gamma_k Equiv$. For a subwalk γ of length s, $Check\gamma_k Equiv$ is called to check for k-obs equivalence against at most all other subwalks of length s ending in the same node. Thus, the worst-case number of same-length subwalks ending in a common vertex for a uniform graph is B^s. As a result, the worst-case complexity of constructing $\mathcal{G}^{(s)}_{po}$ is given by the required five loops in Algorithms 1 and 2, leading to a complexity of $\frac{s|\mathcal{E}^{(s)}|^2}{|\mathcal{V}|}$. The complexity as a function of s is $\mathcal{O}(s|\mathcal{E}^{(s)}|^2)$.

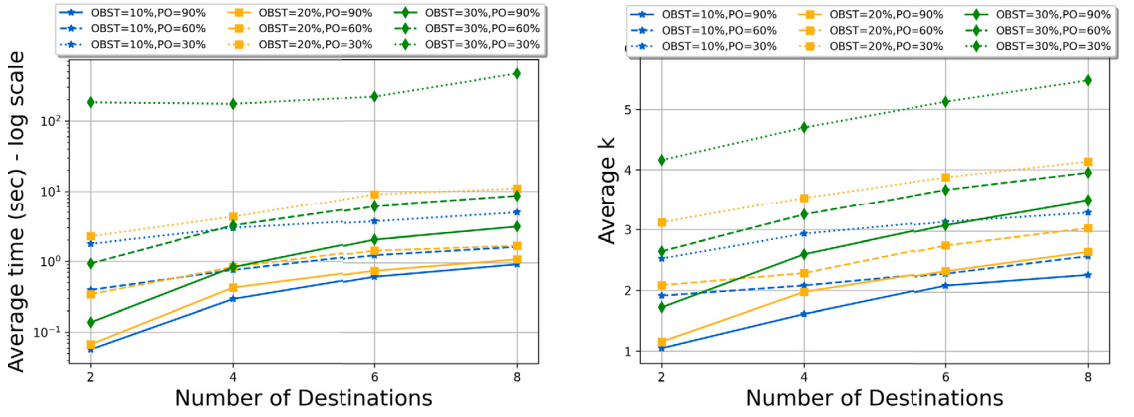
In a full observable setting presented by Bernardini et al. [8], the complexity is $\mathcal{O}(|\mathcal{E}^{(s-1)}|^2)$. Unsurprisingly, the complexity is higher when dealing with partial observability.

9. Experimental results

The experiments were conducted using a cluster with Intel Xeon E5-2640 processors running at 2.60 GHz. The memory limit by process was set to 8 GBs and the time limit to 172,800 seconds.

We use 30×30 2D orthogonal grid graphs. We generate 900-node bidirectional planar graphs with edges connecting each node to its immediate horizontal and vertical neighbors.

To generate harder problems and diversify the graphs, we intervene over two of their configuration features: connectivity and observability. We modify the former by converting 10%, 20%, and 30% of the nodes into obstacles, where an obstacle is a non-passable node in the grid graph (i.e., a node not connected to any other node). This adjustment decreases the graph density and creates bottlenecks. In terms of observability, we generate three new classes of problems by controlling the percentages of observable edges: 90%, 60%, and 30%. The percentage of unobservable edges is the complement of the observable ones, respectively 10%, 40%, and 70%. One bidirectional edge counts as two and if a bidirectional edge is unobservable, it means that moving across the edge in either direction cannot be observed. Both increasing the number of obstacles and decreasing the observability enforce higher



(a) Runtime for calculating the legibility delay.

(b) Change in the legibility delay based on the number of destinations and obstacle and observability ratios.

Fig. 17. Experiments on the legibility delay for a 30×30 grid with destinations varying from 2 to 8, obstacle ratios of 10%, 20%, and 30%, and observability (PO) ratios of 90%, 60%, and 30%.

minimum legibility delays. Further, we split the problems into four more categories based on the number of destinations: from two to eight destinations with 2 unit increment. Note that all the operations for selecting obstacles, unobservable edges, origin, and destination nodes are random, and all destinations are reachable from the origin node. These interventions over the graphs' structures and the number of destinations diversify the sample of problems, which allows us to study the efficiency of our legibility delay calculations under partial observability.

We perform, thus, experiments for thirty-six classes of problems, and we analyze the performance of solving Problems 1, 2, and 3 in relation to three factors: number of destinations, obstacle, and observability (PO) ratio. We run 300 problems per category and report the average values. For each problem instance in a category, we set a different seed.

Fig. 17 displays the results related to Problem 1. Fig. 17(a) shows the average time it takes to calculate the minimum legibility delay for each type of problem. Considering that the runtimes corresponding to the class of problems with obstacle and observability ratio 30% are approximately two to three orders of magnitude higher than the rest, we use the log scale on the time axis to represent the results. The correlation between the average minimum legibility delays and the runtime to calculate them observed in Fig. 17 shows that, as expected, the time complexity of a problem increases proportionally with the minimum legibility delay index.

However, we notice a few apparently unexpected results where this trend does not hold. In particular, we see that for instances with 2 destinations and partial observability percentage of 60%, the average runtime for solving problems with an obstacle ratio of 10% is slightly higher than the one for solving problems with an obstacle ratio of 20%. This phenomenon happens because some cases are easier to analyze than others given how the graphs are structured, regardless of the number of obstacles. In particular, given a legibility delay s, before running the relevant flow algorithms, we always check that, for each destination d in $\mathcal{D}$, there is at least one s-observable walk from the origin to d (see Definition 3). As explained in Section 8, this check can be done efficiently by analyzing the graph $\mathcal{G}_{po}^{(s)}$. If we cannot find any s-observable walk from the origin to d, we already know that the instance cannot be s-legible and avoid running the algorithms altogether. The effect of this optimization can be also observed in the case of problems with 2 destinations, obstacle ratio of 10% and PO of 90%, which have a higher average runtime than problems with 2 destinations, obstacle ratio of 30% and PO of 60%, and, finally, in the case of problems with obstacle ratio of 30% and PO of 30%, when comparing the runtime between problems with 2, 4, and 6 destinations.

Fig. 17(b) represents the average minimum legibility delay calculations. The lines correspond to nine different classes of problems determined by obstacle - observability ratios. The results are averaged out as a function of the number of destinations for each class of problems. As expected, the minimum legibility delay increases as the observability decreases which happens as a result of the increasing number of consecutive unobservable edges. Naturally, the minimum legibility increases with the number of destinations, a behavior also consistent with the findings of Bernardini et al. [8].

We observe that the lines corresponding to PO of 90% present a different trend with respect to the other plot lines. This is due to the average over the problems with 2 destinations being close to 1. Recall that a mandatory condition to find that the minimum legibility delay of an instance is 1 is to have $|D|$ distinct observable edges that leave the origin node. In a graph where the origin node has a maximum connectivity of 4, the probability of finding two observable outgoing edges from o is high, especially if the PO is 90%. On the other hand, for problems with 4 destinations, the probability of having four observable outgoing edges from the origin, thus finding 1-legibility, is reduced (considering also the obstacle percentages). Hence, we have a difference of almost one unit between the average minimum legibility of problems with 2 and 4 destinations.

If we analyze the change in the average minimum legibility when observability decreases, we see that the average minimum legibility delay for the majority of the instances is 4 at the most, even for harder classes of problems such as problems with obstacle ratio of 30%, PO of 60% and 8 destinations, or problems with obstacle ratio of 20%, PO of 30% and 8 destinations. Thus, we can infer

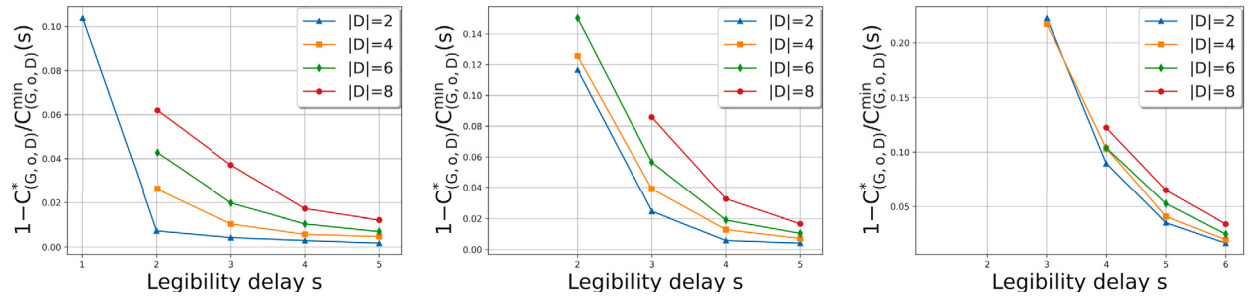


Fig. 18. Experiments on 30×30 grid graphs with obstacle percentage of 20%. The graphs show the change in the budget to reach different legibility delays. The index $1 - C^*_{(\mathcal{G},o,D)}/C^{min}_{(\mathcal{G},o,D)}(s)$ represents the distance from optimality when we constrain the legibility delay.

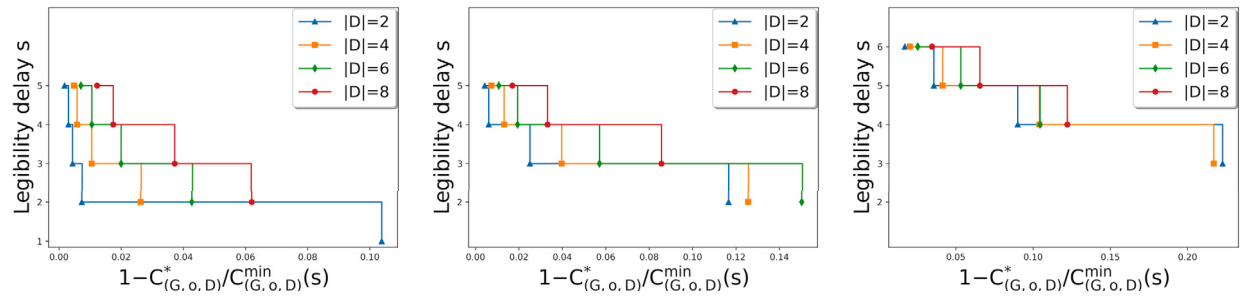


Fig. 19. Experiments on 30×30 grid graphs with obstacle percentage of 20%. The graphs show the change in the legibility delay determined by different budgets. The index $1 - C^*_{(\mathcal{G},o,D)}/C^{min}_{(\mathcal{G},o,D)}(s)$ represents the distance from optimality when we constrain the legibility delay.

that our method is able to successfully find low minimum legibility by searching for walks preponderantly composed of observable edges and minimal overlapping.

We define $C^*_{(\mathcal{G},o,D)}$ as the minimum cost of an (o, D)-connecting set of walks in $\mathcal{G}$. The index $1 - C^*_{(\mathcal{G},o,D)}/C^{min}_{(\mathcal{G},o,D)}(s)$ represents the distance from optimality when we constrain the legibility delay. We use this index to assess performance for Problems 2 and 3 on graphs with obstacle percentages of 20% and PO of 90%, 60%, and 30%.

Fig. 18 relates to Problem 2 in which we evaluate the distance from optimality as a function of the legibility delay index. We observe that the slopes become steeper as the observability decreases. This phenomenon is consistent with our previous conclusion: in order to obtain lower legibility delays, the algorithm searches for walks with a lower number of consecutive unobservable edges which, consequently, become longer as the observability decreases. Thus, $1 - C^*_{(\mathcal{G},o,D)}/C^{min}_{(\mathcal{G},o,D)}(s)$ is higher for the same legibility delay index with respect to the PO ratio of a given graph.

An unexpected behavior is observed in Fig. 18(c) where the lines of problems with more destinations cross under the lines of problems with fewer destinations. This situation can be observed in the case of problems with 2 and 4 destinations for legibility delay 2, and in the case of problems with 4 and 6 destinations for legibility delay 4. This behavior is easily explainable. Let us consider a problem with 4 destinations and one with 2 destinations with the same seed. If, for the extra two destinations, the algorithm finds the optimal walks for a given s-legibility, the difference between $C^{min}_{(\mathcal{G},o,D)}(s)$ and $C^*_{(\mathcal{G},o,D)}$ is the same for both problems. However, proportionally the distance from optimality will be lower in the case where $C^*_{(\mathcal{G},o,D)}$ is higher. Hence, the distance from optimality of the problem with 4 destinations is lower than the one of the problem with 2 destinations.

Fig. 19 focuses on Problem 3 in which we assess the change of the legibility delay s as a function of the distance from the optimality index. As expected, a bigger budget allows us to reach a lower legibility delay. As the observability decreases both the budget and the corresponding legibility delay increase. This is a natural consequence of the increased number of consecutive unobservable edges, which results in finding longer walks with lower legibility as the PO decreases.

10. Conclusions

We explore the concept of goal legibility and recognition in a path planning scenario where the actors' movements are only partially observable to the observer in the loop. In line with previous work on legibility in fully observable settings [8], we present and solve three optimization problems by which we study how to achieve minimum legibility delay for the observer and

how the legibility delay is influenced by the actors' available budget, in addition to the degree of observability of the environment.

We show that those problems can be transformed into classical optimization network flow problems when we make suitable transformations on the graph underlying the pathfinding problem. However, because of partial observability, this transformation is much more sophisticated than in the case of full observability. We present an in-depth theory for this case and prove that the transformation is sound. Our experiments show the viability of our approach.

Based on the foundational study presented here, in future work, we plan to investigate more fine-grained definitions of goal legibility, to study how the construction of legible paths is impacted by dynamic obstacles and partial fields of view and to port the concepts and algorithms we developed for path planning to task planning.

CRediT authorship contribution statement

Sara Bernardini: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **Fabio Fagnani:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Alexandra Neacsu:** Visualization, Software. **Santiago Franco:** Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We thank the anonymous reviewers and the associate editor for their detailed and rigorous reviews, which allowed us to greatly improve the paper.

This work has been partially supported by EPSRC Grant EP/S016473/1 and Leverhulme Trust Grant VP1-2019-037. It has also been partially funded by the European Union Next-GenerationEU grant (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.3 – D.D. 1555 11/10/2022, PE00000013) within the FAIR (Future Artificial Intelligence Research) program.

This manuscript reflects only the authors' views and opinions; none of the funding agencies can be considered responsible for them.

Appendix A. Full construction of $\mathcal{G}_{po}^{(s)}$ for the graph in Fig. A.20

In this section, we present step-by-step the iterative process of finding the minimum legibility delay for the graph $\mathcal{G}$ in Fig. A.20, with a focus on the construction of $\mathcal{G}_{po}^{(s)}$ for $s = 2, 3, 4$. In $\mathcal{G}$, o is the origin node and $\mathcal{D} = \{d_1, d_2, d_3\}$ is the set of destinations. The observable edges are represented with a solid line, and the set of unobservable edges $\mathcal{E}_{nobs} = \{z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8\}$ are marked with a dashed line. In the figures displaying $\mathcal{G}_{po}^{(s)}$, the subgraphs that do not lead to any destination are represented in gray.

We can easily see that 1-legibility cannot be achieved as we cannot find three fully observable edge-distinct walks in $\mathcal{G}$ towards each of the destinations $d \in \mathcal{D}$. By applying the algorithm for 1-legibility, the graph $\mathcal{G}$ under partial observability converts into a fully observable setting by removing all edges in $\mathcal{E}_{nobs}$. This operation leaves d_1 and d_2 unreachable from the origin. Therefore, at step $s = 1$, we do not proceed with calculating the maximum flow, and we increment s to 2.

Fig. A.21 represents the construction of $\mathcal{G}_{po}^{(2)}$ in which only $\bar{d}_3$ is reachable from $\bar{o}$ by applying the first three rules in Expression (9). Given that the walk towards $\bar{d}_3$ is fully observable, such a walk contains only subwalks from $\mathcal{Q}^{(1)\dashv}_{\equiv_{0-obs}}$. From Expression (7) and according to the second rule in Expression (9), the nodes given by the subwalks $[(\alpha_8)]_0$, $[(\alpha_3)]_0$, $[(\alpha_5)]_0$ and $[(\alpha_7)]_0$ are the successors of $\bar{o}$. Since α_3, α_5 and α_7 are followed by one or more unobservable edges in $\mathcal{G}$, we introduce auxiliary constructions. By applying Expression (11), we obtain the auxiliary constructions: $[(\alpha_3)]_0 \to [(\alpha_3)]'_1 \to [(\alpha_3)]_1$, $[(\alpha_5)]_0 \to [(\alpha_5)]'_1 \to [(\alpha_5)]_1$ and $[(\alpha_7)]_0 \to [(\alpha_7)]'_1 \to [(\alpha_7)]_1$. The fifth rule in Expression (9) is then applied to the outgoing edges of $[(\alpha_3)]_1$, $[(\alpha_5)]_1$ and $[(\alpha_7)]_1$, resulting in the subgraphs: $[(\alpha_3)]_1 \xrightarrow{[(\alpha_3 z_3)]_1} [(z_3)]_0$, $[(\alpha_3)]_1 \xrightarrow{[(\alpha_3 z_2)]_1} [(z_2)]_0$, $[(\alpha_5)]_1 \xrightarrow{[(\alpha_5 z_5)]_1} [(z_5)]_0$ and $[(\alpha_7)]_1 \xrightarrow{[(\alpha_7 z_7)]_1} [(z_7)]_0$. According to the first rule in Expression (9), we extend $[(z_5)]_0$ to $[(\alpha_3)]_0$ through the edge $[(z_5 \alpha_3)]_0$.

As seen in Example 7, it is impossible to encode two consecutive unobservable edges in a $\mathcal{G}_{po}^{(2)}$ since a walk containing such a subwalk would not satisfy condition (i) of Definition 4. Therefore, the nodes $[(z_2)]_0, [(z_3)]_0, [(z_7)]_0$ cannot be extended further as the only rules that would involve outgoing edges ending in unobservable edges in Expression (9) cannot be applied in these situations.

Notice that $\mathcal{G}_{po}^{(2)}$ does not contain $\bar{d}_1$ and $\bar{d}_2$. According to the optimization introduced in Section 8, the check for 2-legibility is omitted since the maximum necessary flow cannot be obtained in the associated flow network, and we increment to $s = 3$.

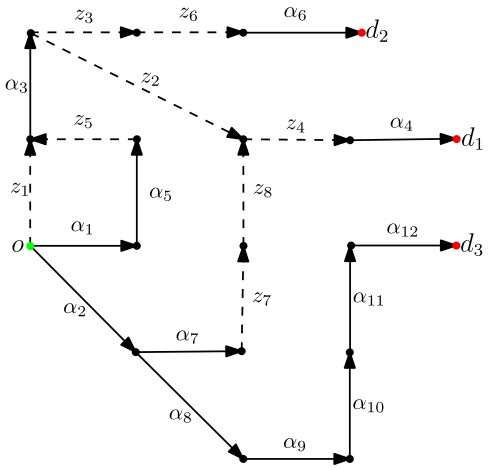


Fig. A.20. A graph $\mathcal{G}$.

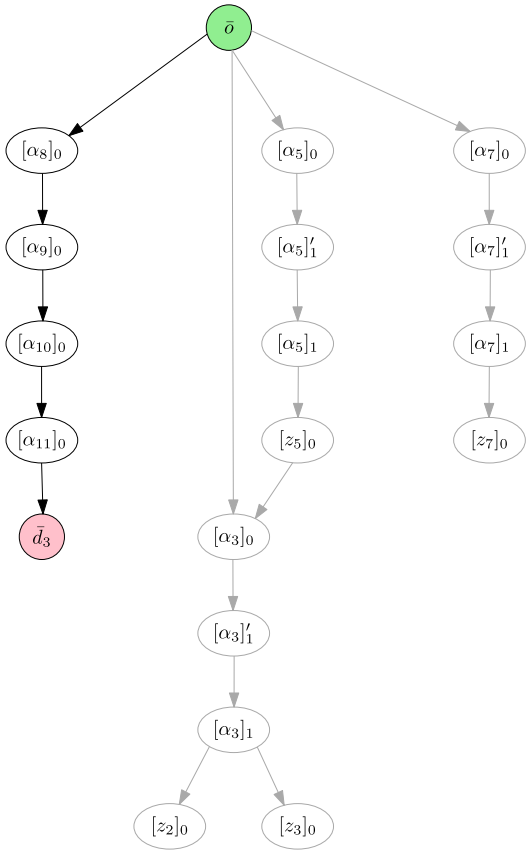


Fig. A.21. $\mathcal{G}^{(2)}$.

There is at least one walk terminating in each destination in D with a maximum of two consecutive unobservable edges, which means that, in this iteration, we can test $\mathcal{G}_{po}^{(3)}$ for 1-legibility.

Note that the walk $(\alpha_2\alpha_7 z_7 z_8 z_4 \alpha_4)$ terminating in d_1 is not 3-observable, and it cannot be fully encoded in $\mathcal{G}_{po}^{(3)}$. Before diving into the construction of $\mathcal{G}_{po}^{(3)}$, visualized in Fig. A.22, let us assess whether $\mathcal{G}$ is 3-legible. There are two other walks leading to d_1 that are 2-observable: $(z_1\alpha_3 z_2 z_4 \alpha_4)$ and $(\alpha_1\alpha_5 z_5 \alpha_3 z_2 z_4 \alpha_4)$. Note that both contain a subwalk of length 4 of form $(null\ \alpha_3\ null\ null)$. Similarly, the two walks leading to destination d_2 contain a subwalk of type $(null\ \alpha_3\ null\ null)$. Any two walks leading to d_1 and d_2 containing the aforementioned subwalk of length 4, whose observable equivalent subwalk is (α_3), cannot be 3-legible nor 4-legible.

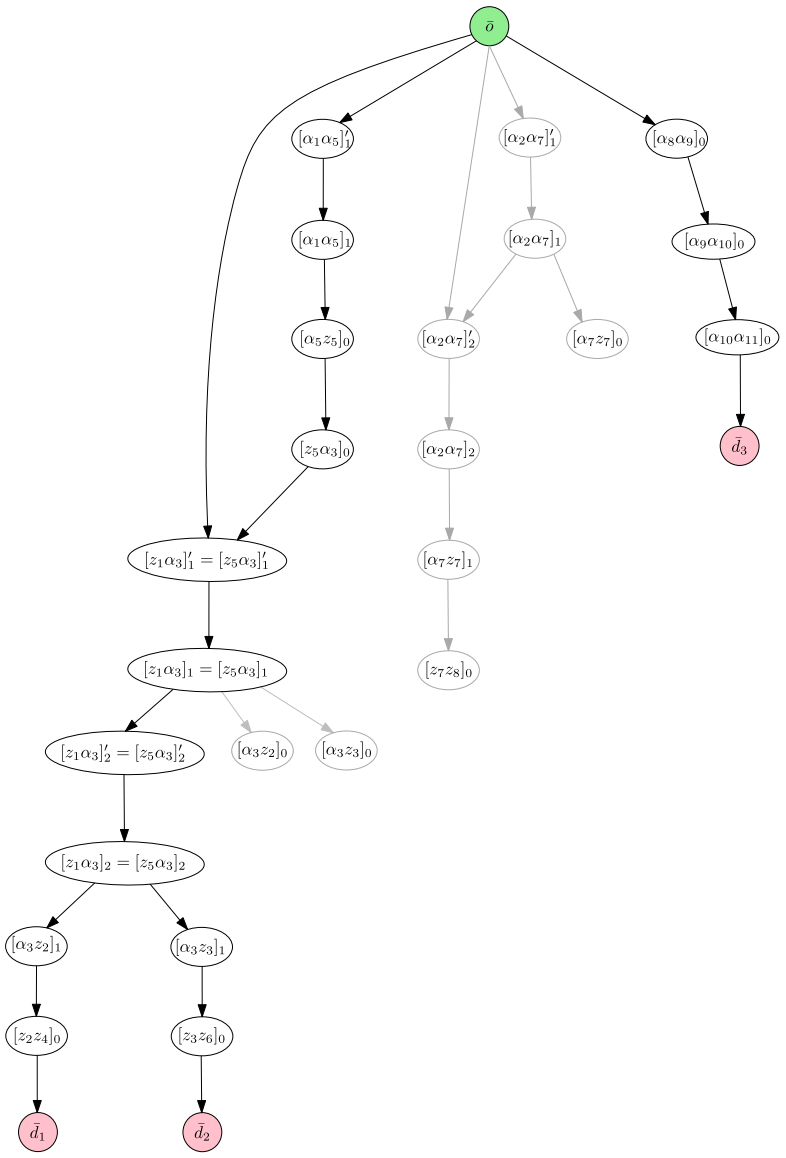


Fig. A.22. $\mathcal{G}^{(3)}$.

Our method encapsulates this structure using collapsed nodes to represent k-obs equivalences. By applying Expression (10) for the subwalk $(z_1\alpha_3)$, we obtain the node-succession $\bar{o} \to [(z_1\alpha_3)]'_1 \to [(z_1\alpha_3)]_1 \to [(z_1\alpha_3)]'_2 \to [(z_1\alpha_3)]_2$, and, by applying Expression (11) for $[(z_5\alpha_3)]_0$, we obtain $[(z_5\alpha_3)]_0 \to [(z_5\alpha_3)]'_1 \to [(z_5\alpha_3)]_1 \to [(z_5\alpha_3)]'_2 \to [(z_5\alpha_3)]_2$. Since α_3 is followed by two consecutive unobservable edges, the auxiliary constructions need to expand up to 2-obs classes of equivalence. Now, we apply the equivalence relation of strong observability described in Definition 7, resulting in the merging of the two walks in $\mathcal{G}^{(3)}_{po}$: $[(z_1\alpha_3)]'_1 = [(z_5\alpha_3)]'_1 \to [(z_1\alpha_3)]_1 = [(z_5\alpha_3)]_1 \to [(z_1\alpha_3)'_2 = [(z_5\alpha_3)]'_2 \to [(z_1\alpha_3)]_2 = [(z_5\alpha_3)]_2$. Collapsing the nodes (thus, creating a bottleneck) guarantees the correct representation of the legibility delay of the corresponding subwalks from $\mathcal{G}$. Therefore, the maximum flow that can be obtained in the flow network corresponding to $\mathcal{G}^{(3)}_{po}$ is two, which means that $\mathcal{G}^{(3)}_{po}$ is not 1-legible.

We further increment s to 4 and obtain $\mathcal{G}^{(4)}_{po}$ displayed in Fig. A.23. Note that, at this stage, the entire partially observable graph can be encoded in $\mathcal{G}^{(4)}_{po}$ since the longest sequence of unobservable edges has length 3. It is easy to observe now that $\mathcal{G}$ is 4-legible since we can reach destination d_1 through a separate walk $(\alpha_2\alpha_7 z_7 z_8 z_4\alpha_4)$, leaving the subwalk of form (*null* α_3 *null null*) to be used to reach d_2.

It is interesting to note how the strongly observable equivalence enunciated in Definition 7 correctly encodes the subgraph created by the subwalks (z_1) and $(\alpha_1\alpha_5 z_5)$ originating in o and terminating in a common node. Starting in $\bar{o}$, the following sequences of nodes are obtained: $\bar{o} \to [(z_1\alpha_3)]'_1 \to [(z_1\alpha_3)]_1 \to [(z_1\alpha_3)]'_2 \to [(z_1\alpha_3)]_2$ (by applying Expression (10)), and $\bar{o} \to [(\alpha_5 z_5\alpha_3)]_0 \to [(\alpha_5 z_5\alpha_3)]'_1 \to [(\alpha_5 z_5\alpha_3)]_1 \to [(\alpha_5 z_5\alpha_3)]'_2 \to [(\alpha_5 z_5\alpha_3)]_2$ (by applying the second rule in Expression (9) and then Expression (11)). Note that the

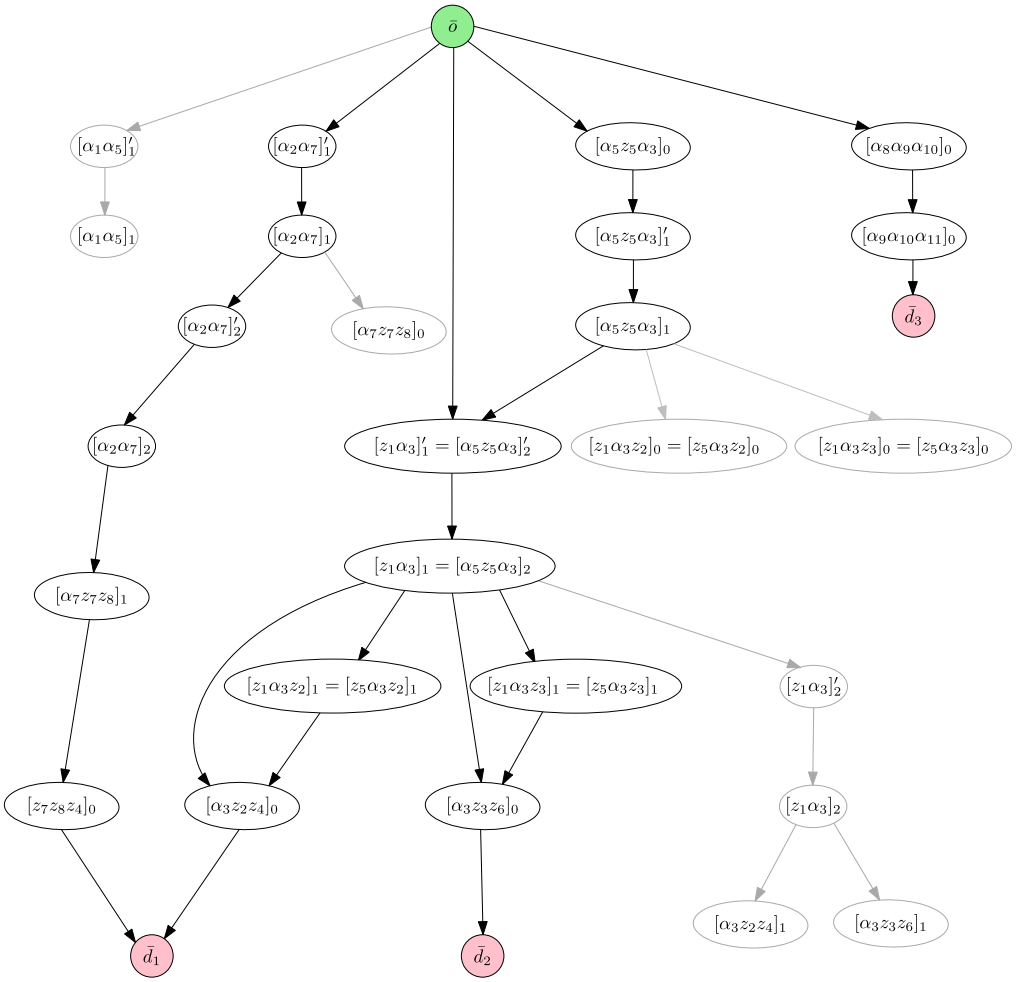


Fig. A.23. $\mathcal{G}^{(4)}$.

walk $(z_1\alpha_3)$ is strongly 0-obs (and 1-obs) equivalent with the suffix of length 2 of walk $(\alpha_5 z_5\alpha_3)$. We then can merge the nodes corresponding to these suffixes to correctly encode their legibility in $\mathcal{G}^{(4)}$, which leads to the following sequence of collapsed nodes: $[(z_1\alpha_3)]'_1 = [(\alpha_5 z_5\alpha_3)]'_2 \rightarrow [(z_1\alpha_3)]_1 = [(\alpha_5 z_5\alpha_3)]_2$.

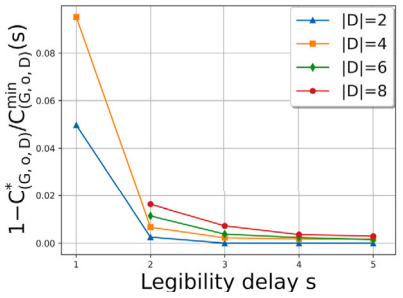
In Fig. A.23, it is visible that the maximum flow in the flow network corresponding to $\mathcal{G}^{(4)}_{po}$ is three and, thus, the minimum legibility delay of $\mathcal{G}$ is 4.

Appendix B. Additional experimental results

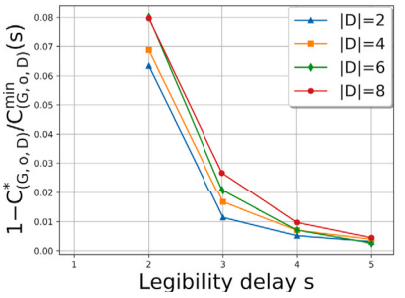
Figs. B.24 and B.25 contain all results related to Problem 2 and Problem 3, respectively. Fig. B.24 displays the change in budget to reach different legibility delays, whereas, Fig. B.25 shows the change in the legibility delay determined by different budgets for all twenty-eight classes of problems.

By analyzing the two figures, we can observe one more trend related to the impact of the obstacle ratio on the results. Namely, the plot lines become more dispersed with the increase in the obstacle percentage. This phenomenon shows that, for Problem 2, for the same legibility delay, generally the distance from optimality increases with the number of destinations more significantly in less connected graphs, unless we can find the optimal walks to the newly added destinations for lower legibility delays, in which case, the distance from optimality improves for problems with more destinations. In the case of Problem 3, the aforementioned effect indicates that, for less connected graphs, the budget allocation needs to increase more significantly when minimizing the legibility delay for problems with more destinations.

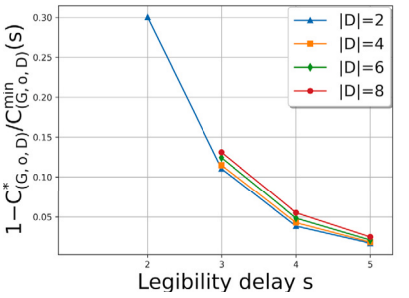
However, we notice that a factor that counters the effect of the obstacle ratio is the decrease in observability. Thus, the difference in performance between problems with different numbers of destinations is less significant by making the problems with lower destinations more expensive cost-wise.



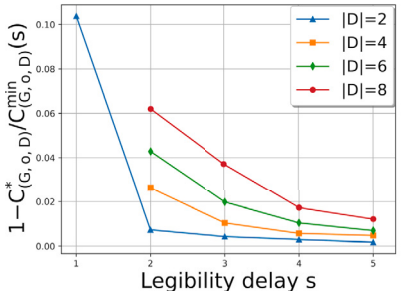
(a) Obstacles 10%; PO 90%



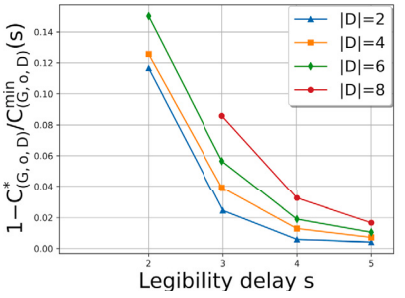
(b) Obstacles 10%; PO 60%



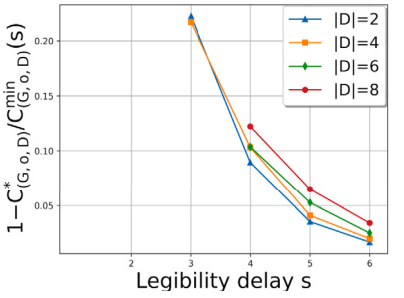
(c) Obstacles 10%; PO 30%



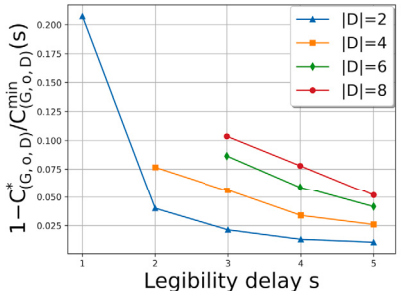
(d) Obstacles 20%; PO 90%



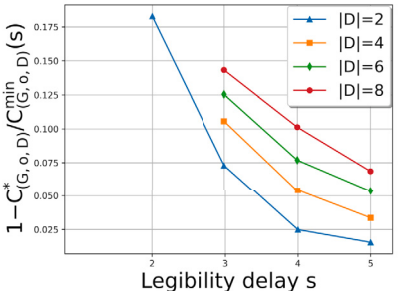
(e) Obstacles 20%; PO 60%



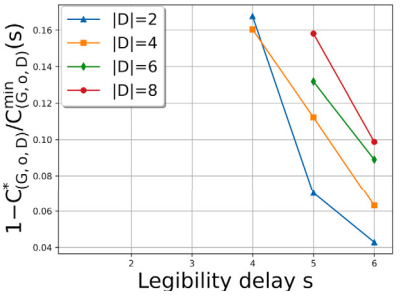
(f) Obstacles 20%; PO 30%



(g) Obstacles 30%; PO 90%



(h) Obstacles 30%; PO 60%



(i) Obstacles 30%; PO 30%

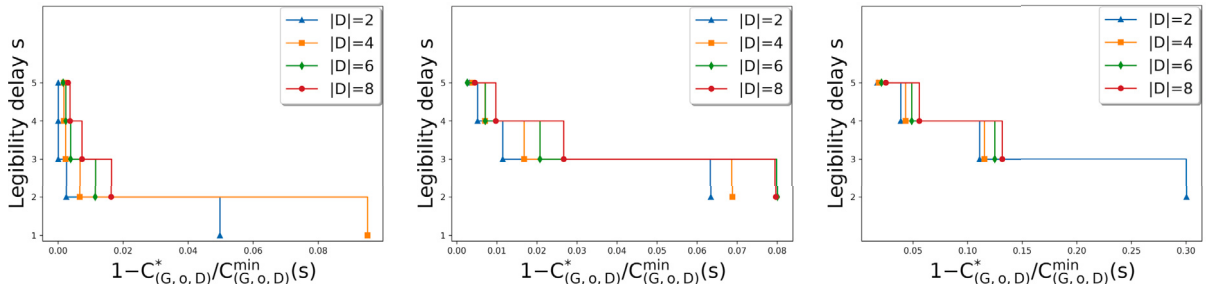
Fig. B.24. Experiments related to Problem 2 on 30×30 grid graphs with destinations varying from 2 to 8, obstacle ratios of 10%, 20%, 30%, and PO ratios of 90%, 60%, and 30%.

List of abbreviations

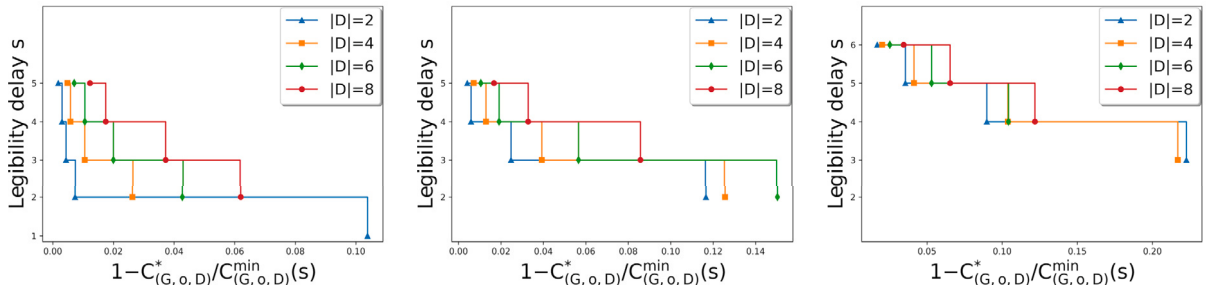
PO	Partially Observable/Partial Observability
PO-LP	Legibility Problem under Partially Observable

List of symbols

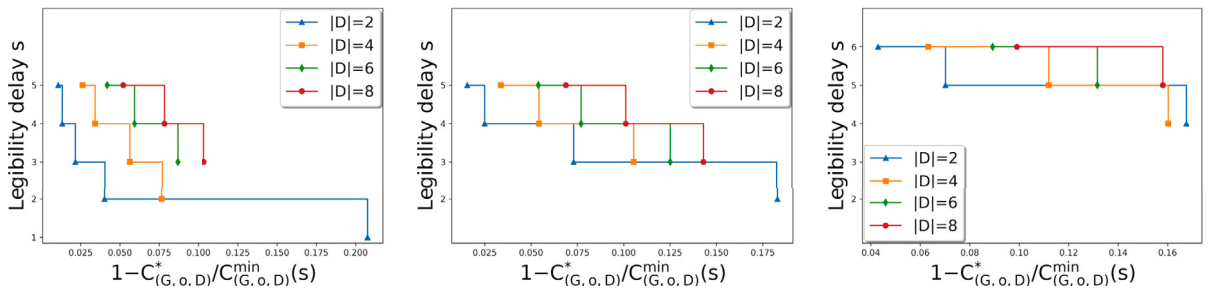
$(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$	PO-LP instance
$LD(\mathcal{P})$	Legibility delay of a set of walks $\mathcal{P}$
$LD(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$	Legibility delay of the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$
$C(\mathcal{P})$	Cost of the set of walks $\mathcal{P}$
$C^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})}(k)$	Minimum cost to achieve k-legibility in the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$
$k^{min}_{(\mathcal{G},\mathcal{E}_{nobs},o,\mathcal{D})}(B)$	Minimum legibility delay achievable under budget B for the PO-LP instance $(\mathcal{G}, \mathcal{E}_{nobs}, o, \mathcal{D})$
$Q^{(s)}$	Set of walks of length s
$\mathcal{I}^{(s)}$	Set of observation sequences of length s
$\Omega^{(s)}$	The function associating sequences of observations of a given length s with destinations



(a) Obstacles 10%; PO 90% (b) Obstacles 10%; PO 60% (c) Obstacles 10%; PO 30%



(d) Obstacles 20%; PO 90% (e) Obstacles 20%; PO 60% (f) Obstacles 20%; PO 30%



(g) Obstacles 30%; PO 90% (h) Obstacles 30%; PO 60% (i) Obstacles 30%; PO 30%

Fig. B.25. Experiments related to Problem 3 on 30×30 grid graphs with destinations varying from 2 to 8, obstacle ratios of 10%, 20%, 30%, and PO ratios of 90%, 60%, and 30%.

$\equiv_{0-obs}$	Equivalence relation between two walks, detailed in Definition 7
$\equiv_{k-obs}$	Equivalence relation between the k suffixes of two walks, detailed in Definition 8
$\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}$ such that every two walks are $\equiv_{k-obs}$
$\mathcal{Q}^{(s)}_{(o\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that start in the origin o but do not end in any destination
$\mathcal{Q}^{(s)}_{d)\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that end in the destination d
$\mathcal{Q}^{(s)}_{(o,d)\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that start in the origin o and end in the destination d
$\mathcal{Q}^{(s)}_{\bullet\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that do not start in the origin and do not end in any destination
$\mathcal{Q}^{(s)\dashv}_{\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that end with an observable edge
$\mathcal{Q}^{(s)\not\dashv}_{\equiv_{k-obs}}$	Equivalence classes of walks from $\mathcal{Q}^{(s)}_{\equiv_{k-obs}}$ that end with an unobservable edge
$\mathcal{G}^{(s)}_{po} = (\mathcal{V}^{(s)}, \mathcal{E}^{(s)}_{bas} \cup \mathcal{E}^{(s)}_{aux})$	s-Legibility graph under partial observability
$\mathcal{E}^{(s)}_{bas}$	Basic edges derived from walks of length s in $\mathcal{G}^{(s)}_{po}$
$\mathcal{E}^{(s)}_{aux}$	Set of auxiliary edges in $\mathcal{G}^{(s)}_{po}$
$W^{(s)}(\omega)$	Weight of a walk ω in $\mathcal{G}^{(s)}_{po}$
$C^{(s)}(\mathcal{R})$	Cost of a set of walks $\mathcal{R}$ in $\mathcal{G}^{(s)}_{po}$

References

[1] J. Alexandersson, Plan recognition in verbmobil, in: Proceedings of IJCAI-95 Workshop the Next Generation of Plan Recognition Systems: Challenges for and Insight from Related Areas of AI (Working Notes), 1995, pp. 2–7.

[2] N. Alon, L. Schulz, J.S. Rosenschein, P. Dayan, A (dis-)information theory of revealed and unrevealed preferences: emerging deception and skepticism via theory of mind, Open Mind 7 (2023) 608–624, https://doi.org/10.1162/opmi_a_00097.

[3] S. Ang, H. Chan, A.X. Jiang, W. Yeoh, Game-theoretic goal recognition models with applications to security domains, in: Decision and Game Theory for Security: 8th International Conference, Proceedings, GameSec 2017, Vienna, Austria, October 23-25, 2017, Springer, 2017, pp. 256–272.

[4] T.C. Au, Extended goal recognition design with first-order computation tree logic, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 9661–9668.

[5] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, P. Havinga, Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: a survey, in: 23th International Conference on Architecture of Computing Systems 2010, VDE, 2010, pp. 1–10.

[6] C.L. Baker, R. Saxe, J.B. Tenenbaum, Action understanding as inverse planning, Cognition 113 (2009) 329–349.

[7] S. Bernardini, F. Fagnani, S. Franco, An optimization approach to robust goal obfuscation, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, 2020, pp. 119–129.

[8] S. Bernardini, F. Fagnani, S. Franco, A. Neacsu, A network flow interpretation of robust goal legibility in path finding, in: Proc. of the 32nd International Conference on Automated Planning and Scheduling, ICAPS 2022, Singapore, 2022, pp. 668–677.

[9] U. Bhatt, P. Ravikumar, J.M.F. Moura, Building human-machine trust via interpretability, Proc. AAAI Conf. Artif. Intell. 33 (2019) 9919–9920.

[10] N. Blaylock, J. Allen, et al., Corpus-based, statistical goal recognition, in: International Joint Conference on Artificial Intelligence, Lawrence Erlbaum Associates Ltd., 2003, pp. 1303–1308.

[11] A. Bobu, A. Peng, P. Agrawal, J. Shah, A.D. Dragan, Aligning robot and human representations, arXiv preprint, arXiv:2302.01928, 2023.

[12] T. Chakraborti, A. Kulkarni, S. Sreedharan, D.E. Smith, S. Kambhampati, 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, 2019, pp. 86–96.

[13] K. Chandra, T.M. Li, J. Tenenbaum, J. Ragan-Kelley, Acting as inverse inverse planning, in: ACM SIGGRAPH 2023 Conference Proceedings, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1–12.

[14] G. Csibra, G. Gergely, 'Obsessed with goals': functions and mechanisms of teleological interpretation of actions in humans, Acta Psychol. 124 (2007) 60–78.

[15] A. Dragan, S. Srinivasa, Integrating human observer inferences into robot motion planning, Auton. Robots 37 (2014) 351–368.

[16] A.D. Dragan, Robot planning with mathematical models of human state and action, CoRR, arXiv:1705.04226, http://arxiv.org/abs/1705.04226, 2017.

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

[18] J.F. Fisac, M.A. Gates, J.B. Hamrick, C. Liu, D. Hadfield-Menell, M. Palaniappan, D. Malik, S.S. Sastry, T.L. Griffiths, A.D. Dragan, Pragmatic-pedagogic value alignment, in: N.M. Amato, G. Hager, S. Thomas, M. Torres-Torriti (Eds.), Robotics Research, Springer International Publishing, Cham, 2020, pp. 49–57.

[19] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, Can. J. Math. 8 (1956) 399–404.

[20] K.C. Gall, W. Ruml, S. Keren, Active goal recognition design, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, 2021, pp. 4062–4068, https://doi.org/10.24963/ijcai.2021/559, ijcai.org.

[21] C.W. Geib, R.P. Goldman, A probabilistic plan recognition algorithm based on plan tree grammars, Artif. Intell. 173 (2009) 1101–1132.

[22] D. Gunning, D. Aha, Darpa's explainable artificial intelligence (xai) program, AI Mag. 40 (2019) 44–58.

[23] C. Heinze, S. Goss, A. Pearce, Plan recognition in military simulation: incorporating machine learning with intelligent agents, in: Proceedings of IJCAI-99 Workshop on Team Behaviour and Plan Recognition, 1999, pp. 53–64.

[24] J. Hong, Goal recognition through goal graph analysis, J. Artif. Intell. Res. 15 (2001) 1–30.

[25] S. Keren, A. Gal, E. Karpas, Goal recognition design in deterministic environments, J. Artif. Intell. Res. (JAIR) 65 (2019).

[26] A. Kulkarni, S. Srivastava, S. Kambhampati, A unified framework for planning in adversarial and cooperative environments, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 2479–2487.

[27] A. Kulkarni, S. Srivastava, S. Kambhampati, Signaling friends and head-faking enemies simultaneously: balancing goal obfuscation and goal legibility, in: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 1889–1891.

[28] P. Langley, B. Meadows, M. Sridharan, D. Choi, Explainable agency for intelligent autonomous systems, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, 2017, pp. 4762–4763.

[29] N. Le Guillarme, A.I. Mouaddib, X. Lerouvreur, S. Gatepaille, A generative game-theoretic framework for adversarial plan recognition, in: Distributed and Multi-Agent Planning (DMAP-15), 2015, pp. 33–41.

[30] A.M. MacNally, N. Lipovetzky, M. Ramirez, A.R. Pearce, Action selection for transparent planning, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multi-Agent Systems, Richland, SC, 2018, pp. 1327–1335.

[31] P. Masters, S. Sardina, Cost-based goal recognition for path-planning, in: Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems, 2017, pp. 750–758.

[32] P. Masters, S. Sardina, Deceptive path-planning, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 4368–4375, https://doi.org/10.24963/ijcai.2017/610.

[33] P. Masters, S. Sardina, Cost-based goal recognition in navigational domains, J. Artif. Intell. Res. 64 (2019) 197–242.

[34] P. Masters, S. Sardina, Expecting the unexpected: goal recognition for rational and irrational agents, Artif. Intell. 297 (2021) 103490.

[35] F.R. Meneguzzi, R.F. Pereira, A survey on goal recognition as planning, in: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), 2021, Canada, 2021, pp. 4524–4532.

[36] W. Min, E. Ha, J. Rowe, B. Mott, J. Lester, Deep learning-based goal recognition in open-ended digital games, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2014, pp. 37–43.

[37] W. Min, B.W. Mott, J.P. Rowe, B. Liu, J.C. Lester, Player goal recognition in open-world digital games with long short-term memory networks, in: IJCAI, 2016, pp. 2590–2596.

[38] S. Miura, A.L. Cohen, S. Zilberstein, Maximizing legibility in stochastic environments, in: 2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN), 2021, pp. 1053–1059.

[39] S. Miura, S. Zilberstein, A unifying framework for observer-aware planning and its complexity, in: 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021), 2021, pp. 610–620.

[40] B. Mott, S. Lee, J. Lester, Probabilistic goal recognition in interactive narrative environments, in: Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA, 1999, AAAI Press/MIT Press, Cambridge, MA/London, 2006, p. 187.

[41] R. Nakahashi, Human-Agent Teaming with Implicit Guidance, Ph.D. thesis, The Graduate University for Advanced Studies, 2022.

[42] R. Pereira, N. Oren, F. Meneguzzi, Landmark-based heuristics for goal recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 3622–3628.

[43] M. Ramırez, H. Geffner, Plan recognition as planning, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc, Citeseer, 2009, pp. 1778–1783.
[44] M. Ramírez, H. Geffner, Probabilistic plan recognition using off-the-shelf classical planners, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2010, pp. 1121–1126.
[45] M. Ramirez, M. Papasimeon, N. Lipovetzky, L. Benke, T. Miller, A.R. Pearce, E. Scala, M. Zamani, Integrated hybrid planning and programmed control for real time uav maneuvering, in: AAMAS, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2018, pp. 1318–1326.
[46] P.C. Roy, A. Bouzouane, S. Giroux, B. Bouchard, Possibilistic activity recognition in smart homes for cognitively impaired people, Appl. Artif. Intell. 25 (2011) 883–926.
[47] S. Sreedharan, A. Kulkarni, S. Kambhampati, Legible Behavior, Springer International Publishing, Cham, 2022, pp. 47–57, Chapter 4.
[48] S. Sreedharan, A. Kulkarni, D.E. Smith, S. Kambhampati, A unifying Bayesian formulation of measures of interpretability in human-ai interaction, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21), 2021, pp. 4602–4610.
[49] R. Stern, Multi-Agent Path Finding–an Overview, Springer, 2019, pp. 96–115, Chapter 1.
[50] D. Strouse, M. Kleiman-Weiner, J. Tenenbaum, M. Botvinick, D.J. Schwab, Learning to share and hide intentions using information regularization, in: Advances in Neural Information Processing Systems, 2018, pp. 10270–10281.
[51] C. Wayllace, P. Hou, W. Yeoh, New metrics and algorithms for stochastic goal recognition design problems, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 4455–4462.
[52] R. Wilensky, Planning and Understanding: A Computational Approach to Human Reasoning, Addison-Wesley Pub. Co., Reading, MA, 1983.
[53] J. Yin, X. Chai, Q. Yang, High-level goal recognition in a wireless lan, in: Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA, AAAI Press/MIT Press, Cambridge, MA/London, 2004, pp. 578–584.
[54] J. Yu, S.M. LaValle, Multi-agent path planning and network flow, in: E. Frazzoli, T. Lozano-Perez, N. Roy, D. Rus (Eds.), Algorithmic Foundations of Robotics X, Springer, Berlin, Heidelberg, 2013, pp. 157–173.
[55] T. Zhi-Xuan, J.L. Mann, T. Silver, J.B. Tenenbaum, V.K. Mansinghka, Online Bayesian goal inference for boundedly-rational planning agents, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2020, pp. 19238–19250.