Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling (ICAPS 2022)

A Network Flow Interpretation of Robust Goal Legibility in Path Finding

Sara Bernardini,[1] Fabio Fagnani, [2] Santiago Franco [1] Alexandra Neacsu,[1]

[1] Department of Computer Science, Royal Holloway University of London, Egham, UK
[2] Department of Mathematical Sciences, Politecnico di Torino, Torino, Italia, 10129
{sara.bernardini, santiago.franco, alexandra.neacsu}@rhul.ac.uk, fabio.fagnani@polito.it

Abstract

In this paper, we define goal legibility in a multi-agent path-finding setting. We consider a set of identical agents moving in an environment and tasked with reaching a set of locations that need to be serviced. An observer monitors their movements from a distance to identify their destinations as soon as possible. Our algorithm constructs a set of paths for the agents, one to each destination, that overlap as little as possible while satisfying a budget constraint. In this way, the observer, knowing the possible agents' destinations as well as the set of paths they might follow, is guaranteed to determine with certainty an agent's destination by looking at the shortest possible fragment of the agent's trajectory, regardless of when it starts observing. Our technique is robust because the observer's inference mechanism requires no coordination with the agents' motions. By reformulating legible path planning into a classical minimum cost flow problem, we can leverage powerful tools from combinatorial optimization, obtaining fast and scalable algorithms. We present experiments that show the benefits offered by our approach.

1 Introduction

In this paper, we focus on *legibility*, which refers to the agent's ability to signal the goal it wants to achieve through its actions to an observer in the loop. The notion of legibility has been introduced in robotics motion planning by Dragan and Srinivasa (2014), who define legible motion as *"motion that enables an observer to quickly and confidently infer the correct goal"*. This concept affords a broad interpretation and has been formulated in different ways across a growing body of work that has come to include motion and task planning (Dragan 2017; MacNally et al. 2018; Kulkarni, Srivastava, and Kambhampati 2019, 2020), single and multi-agent scenarios (Miura and Zilberstein 2021) and deterministic and stochastic settings (Miura, Cohen, and Zilberstein 2021). Legibility also overlaps with the concept of transparency (MacNally et al. 2018).

We consider a cooperative path-finding setting in an environment with one origin and a set of destinations. One or more identical agents can appear at the origin and move in the environment to serve requests at the different destinations. An observer monitors the environment from a distance. It can see all agents' actions and wants to determine their destinations as quickly as possible after starting observing their motion. The problem is to synthesize paths for the agents so as to make it easy for the observer to infer their destinations. We solve the problem as follows. In an offline phase, a path finder, which knows the origin and the destinations, builds paths from the origin to each destination so that the paths overlap as little as possible. Then, the path finder communicates these paths both to the observer and the agents. In a subsequent online phase, each agent entering the environment picks one destination and follows the pre-calculated path to it. The observer, given how the paths have been constructed, is guaranteed to be able to decide the agents' destinations with certainty by making the minimum possible number of observations.

This setting is motivated by autonomous robotic missions in challenging scenarios, where the agent and the human do not share the same physical space and do not interact directly. Instead, the robots autonomously perform some tasks, and a human supervisor 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 (direct communication might not be available). Our goal is to simplify the decision-making process of the observer and reduce their cognitive load. For example, consider an autonomous firefighting system operating in a factory where some rooms are at risk of fire and need to be periodically inspected. Our technique provides the robots with paths that have minimal overlap so that a human looking at any snippet of an agent's movements (not necessarily starting from the origin) can quickly determine where it is going.

Our first contribution in this paper is to define the concept of legibility within a multi-agent *path finding* setting and give a crisp mathematical formalization of this problem. We propose an *optimization* framework to look for optimal solutions under a budget constraint. Building on this framework, our second contribution is to provide efficient algorithms to solve legible path finding by leveraging classical techniques from combinatorial optimization. We show that our problem can be reformulated in terms of finding the *minimum cost flow* in a suitably constructed *network*. Our approach is *robust* as the observer's reasoning does not need to be coordinated with the agents' actions. In related techniques, the observations are always assumed to start from the first step of the agent's behavior, while in our case the observer can start observing the agent's movement at any point and still be able to decide the agent's destination with complete confidence. Our experiments show that our approach is powerful and can scale up to large environments efficiently.

2 Problem Statement

In general, multi-agent path planning involves three problems: (i) assign agents to destinations; (ii) find a path for each agent to its assigned destination; and (iii) schedule the agents' motions on the paths (Yu and LaValle 2013). Here, we only focus on the second problem, path finding, and, more specifically, on *legible path finding*. The first problem is irrelevant as we assume that the agents are identical so each agent can be associated with any destination (anonymous path planning). As for the third problem, we assume no global coordination between the agents, so they can start their motion at different times. If a local conflict emerges, because two or more agents need to move to the same vertex or swap position, we assume simple local coordination policies can be adopted (e.g. one agent moves, the others wait).

2.1 Legibility Definitions

We now give a formal definition of legibility in multi-agent path planning. We consider directed multigraphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. We implicitly 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 the head of an edge $e \in \mathcal{E}$ respectively. Given a vertex $v \in \mathcal{V}$, we define $in(v)$ and $out(v)$ as follows:

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

A walk from a node v to a node w is a sequence of edges $\gamma = (e_1, \ldots, e_l)$ such that $\theta(e_1) = v$, $\kappa(e_l) = w$, and $\theta(e_h) = \kappa(e_{h-1})$ for all $h = 2, \ldots, l$. For a walk γ, we indicate its length as $l_\gamma = l$. We work with *multigraphs* because the constructions in Sections 3 and 4 become simpler when using this concept, which is more flexible than the concept of graphs. In the examples, when there are not 'parallel' edges, we represent graphs, where edges are represented as ordered pairs of vertices.

A node $o \in \mathcal{V}$ represents the origin node and $\mathcal{D} \subseteq \mathcal{V}$ a set of 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 ($\forall d \in \mathcal{D},\ out(d) = \emptyset$). The triple $(\mathcal{G}, o, \mathcal{D})$ is called a *legibility problem instance*.

Given a set of walks $\mathcal{P}$ from o to $\mathcal{D}$, 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 $(o, \mathcal{D})$-*connecting *if, for every $d \in \mathcal{D}$, there exists a walk $\gamma \in \mathcal{P}$ from o to d (i.e. $\forall d \in \mathcal{D}, \mathcal{P}^{(d)} \neq \emptyset$).*

We now introduce the concept of legibility formally.

Definition 2 *An $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ is called* k-legible*, with $k \in \mathbb{N}^+$, if $\forall\, d, d' \in \mathcal{D}$ with $d \neq d'$, $\forall\, \gamma = (e_1, \ldots, e_{l_\gamma}) \in \mathcal{P}^{(d)}$ and $\forall\, \gamma' = (e'_1, \ldots, e'_{l_{\gamma'}}) \in \mathcal{P}^{(d')}$, we have that $(e_{j+1}, \ldots, e_{j+k}) \neq (e'_{j'+1}, \ldots, e'_{j'+k})$, $\forall j, j' \in \mathbb{N}$ such that $j + k \leq l_\gamma$ and $j' + k \leq l_{\gamma'}$.*

Definition 2 means that the set $\mathcal{P}$ is k-legible if, given any two walks from the origin to two different destinations, they do not share any subwalk of length k. For example, two walks are 1-legible if they do not have any edge in common; they are 2-legible if they do not share any subwalk of length two (but they might have an edge in common). Clearly, if two walks are 1-legible, they are also 2-legible and so on. Hence, we are interested in the minimum legibility.

Definition 3 *Given an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$, the minimum k such that $\mathcal{P}$ is k-legible is called the* legibility delay *of $\mathcal{P}$ and denoted as $k(\mathcal{P})$.*

Definition 3 tells us that if a set of walks $\mathcal{P}$ has legibility delay k, the observer is guaranteed to determine each agent's destination by making k observations at the most. Consider a prefix-free language $\mathcal{L}$ whose words are the subwalks of the walks in $\mathcal{P}$ with length up to k. From a practical point of view, after building the walks in $\mathcal{P}$, we could construct a look-up table to associate words of $\mathcal{L}$ with their corresponding destinations. The observer, looking at this table, would only need to wait until it observes one of the entries to then pick the associated destination as the agent's chosen one.

Considering Definitions 2 and 3, we have the following observation.

Remark 1 *When we check whether a set of walks $\mathcal{P}$ is k-legible, all the walks of length $l \leq k$ do not play any role. This implies that, if we put*

$$l(\mathcal{P}) = \max_{\gamma \in \mathcal{P}} l_\gamma$$

every set $\mathcal{P}$ is $l(\mathcal{P})$-legible.

The following definition concludes our formal introduction of legibility.

Definition 4 *A legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ is called* k-legible *if there exists an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ that is k-legible. The* legibility delay *of $(\mathcal{G}, o, \mathcal{D})$, denoted $k(\mathcal{G}, o, \mathcal{D})$, is the minimum legibility delay among all possible $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$. In formula,*

$$k(\mathcal{G}, o, \mathcal{D}) = \min_{\mathcal{P}\ (o,\mathcal{D})-conn.} k(\mathcal{P})$$

2.2 Legibility Optimization Problems

We now consider the three most relevant optimization problems concerning the concept of legibility.

Problem 1 *Given a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$, determine its legibility delay $k(\mathcal{G}, o, \mathcal{D})$ and find an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ such that $k(\mathcal{P}) = k(\mathcal{G}, o, \mathcal{D})$.*

A simple observation follows. We will use it in Section 3.

Remark 2 *We can always restrict our search to $(o, \mathcal{D})$-connecting sets $\mathcal{P}$ such that $|\mathcal{P}| = |\mathcal{D}|$, i.e. subsets containing exactly one walk from o to every destination d.*

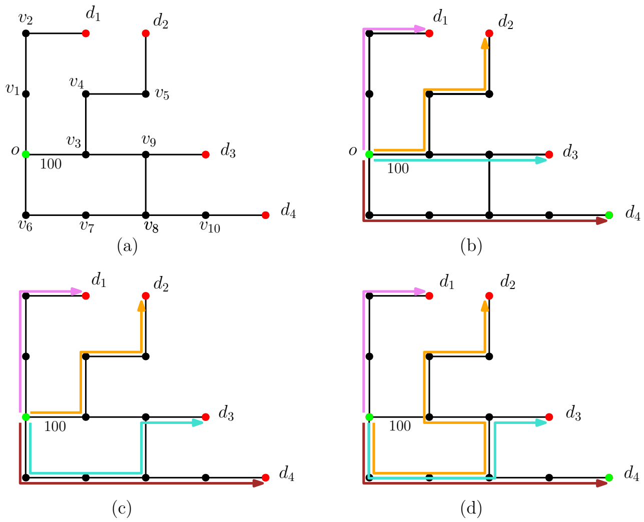


Figure 1: (a): A legibility problem instance $(\mathcal{G}, o, \mathcal{D})$. (b): $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ such that $k(\mathcal{P}) = 2$ for an infinite budget. (c): The legibility delay increases to 4 for a budget $B = 120$. (d): The legibility delay increases even more ($k(\mathcal{P}) = 5$) for a budget $B = 100$.

We now introduce a *cost* on walks in the following way. 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 $W(\gamma) = \sum_{h=1}^{l} W_{e_h}$. Finally, the cost of a set of walks $\mathcal{P}$ is defined as

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

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

Problem 2 *Given a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ and a legibility delay k, find an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ that is k-legible and minimizes the cost $C(\mathcal{P})$.*

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

We conventionally put $C^{min}(k) = \infty$ if $k(\mathcal{G}, o, \mathcal{D}) > k$.

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

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

Example 1 *Consider the legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ as depicted in Figure 1(a). Assume all edges have unitary cost, except for edge (o, v_3), which has cost 100. Suppose that you have an infinite budget first. It is easy to see that $k(\mathcal{G}, o, \mathcal{D}) \geq 2$ because, since the destinations are four but the edges leaving o are only three, two walks will have to share one edge in order to reach all destinations. Figure 1(b) shows an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ such that $k(\mathcal{P}) = 2$. The legibility delay is two as edge (o, v_3) is traversed both to reach d_2 and d_3, so an observer needs to wait at least two steps to decide where the agent is going. As a consequence, $k(\mathcal{G}, o, \mathcal{D}) = 2$. Now, suppose that the available budget is 120 ($B = 120$). In this case, the set of walks $\mathcal{P}$ cannot be utilized to reach the destinations as $C(\mathcal{P}) > B$. Figure 1(c) shows that, to satisfy the budget constraint, we can traverse (o, v_3) only once. This constraint increases the legibility delay, which becomes $k(\mathcal{G}, o, \mathcal{D}) = 4$, as the walks to d_3 and d_4 now share three edges, i.e. $(o, v_6), (v_6, v_7), (v_7, v_8)$. If the budget is even smaller, $B = 100$ (Figure 1(d)), edge (o, v_3) needs to be avoided altogether, increasing the legibility delay further, $k(\mathcal{G}, o, \mathcal{D}) = 5$, as now the walks to d_2 and d_3 share four edges, i.e. $(o, v_6), (v_6, v_7), (v_7, v_8), (v_8, v_9)$.*

3 1-Legibility as a Network Flow Problem

In this section, we study 1-legibility. In particular, we focus on finding 1-legible sets when they exist and solving Problem 2 for the special case $k = 1$. In the next section, we present a method to reduce k-legibility for any k to 1-legibility on a suitably modified graph.

Checking 1-legibility boils down to verify that we can find an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ that do not have any edge in common, i.e. they are edge-independent. We start our analysis with some initial observations.

3.1 Preliminary Observations

If the number of edges that exit the origin is smaller than the number of destinations, then necessarily one or more of those edges will have to be taken more than once in constructing an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$, which violates the conditions for 1-legibility. Hence, the following necessary condition holds for 1-legibility: $|out(o)| \geq |\mathcal{D}|$. The second observation is that, for 1-legibility, we can restrict our analysis to paths (instead of walks) in the graph. As we look for edge-independent walks, being able to go back and forth on them does not impact legibility. We have the following result.

Proposition 1 *If there exists an $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ such that $k(\mathcal{P}) = 1$, then there exists an $(o, \mathcal{D})$-connecting set of paths $\mathcal{P}'$ such that $k(\mathcal{P}') = 1$.*

Proof The elimination of possible closed subwalks from the walks in $\mathcal{P}$ does not impact either $(o, \mathcal{D})$-connectivity or edge-independency. This consideration proves the claim. ■

It is convenient to make a modification to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by creating a new fictitious destination $\bar{d}$, which we add to $\mathcal{V}$. We also add an edge from each destination $d \in \mathcal{D}$ to $\bar{d}$ to the set of edges $\mathcal{E}$. We call this new graph the *modified graph* and indicate it as

$$\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}}) \tag{1}$$

Any set of walks $\mathcal{P}$ from o to $\mathcal{D}$ can be transformed into a set $\bar{\mathcal{P}}$ of walks from o to $\bar{d}$ in $\bar{\mathcal{G}}$ by adding a last link to $\bar{d}$ in each walk. Also, by dropping the last edge from the walks in $\bar{\mathcal{P}}$, we can reconstruct $\mathcal{P}$.

Since we assume that $|\mathcal{P}| = |\mathcal{D}|$ (see Remark 2), it holds that $\mathcal{P}$ is 1-legible if and only if the paths in $\bar{\mathcal{P}}$ are edge-independent. In consequence, our problem of checking if $(\mathcal{G}, o, \mathcal{D})$ is 1-legible is equivalent to checking if there exist $|\mathcal{D}|$ distinct paths in $\bar{\mathcal{G}}$ from o to $\bar{d}$. This is a classical graph theory problem that, based on Menger's theorem (Menger 1927), can be reformulated in terms of cut capacities as follows. An $(o - \bar{d})$-cut (A, B) is a partition of vertices $\bar{\mathcal{V}}$ into subsets A and B such that $o \in A$ and $\bar{d} \in B$. The cut capacity, $q(A, B)$, is the cardinality of the set of edges going from A to B. We define the $(o - \bar{d})$-*connectivity* of $\bar{\mathcal{G}}$ as the minimum cut capacity over all possible $(o - \bar{d})$-cuts:

$$\lambda^*(\bar{\mathcal{G}}, o, \bar{d}) = \min\{q(A, B) \,|\, (A, B)\ (o - \bar{d}) - \text{cut}\}$$

This index can be interpreted as the minimum number of edges whose removal disconnects o from $\bar{d}$. Notice that, since the capacity of the cut $(\bar{\mathcal{V}} \setminus \{\bar{d}\}, \{\bar{d}\})$ is $|\mathcal{D}|$, we always have that $\lambda^*(\bar{\mathcal{G}}, o, \bar{d}) \leq |\mathcal{D}|$.

The following corollary summarizes our analysis of the legibility problem so far.

Corollary 1 *Consider a legibility problem instance* $(\mathcal{G}, o, \mathcal{D})$. *The following conditions are equivalent:*

1. $k(\mathcal{G}, o, \mathcal{D}) = 1$
2. *There exist* $|\mathcal{D}|$ *edge-independent paths in* $\bar{\mathcal{G}}$ *from* o *to* $\bar{d}$.
3. $\lambda^*(\bar{\mathcal{G}}, o, \bar{d}) = |\mathcal{D}|$

Proof The equivalence between 1 and 2 follows from the definition of $\bar{\mathcal{G}}$ in (1). The equivalence between 2 and 3 follows from the Menger's theorem. ■

These considerations reformulate producing an $(o, \mathcal{D})$-connecting 1-legible set of walks in $\mathcal{G}$ into finding the maximum number of independent paths from o to $\bar{d}$ in the modified graph $\bar{\mathcal{G}}$ and, in turn (by the Menger's theorem), into exhibiting a $(o - \bar{d})$-cut with capacity $|\mathcal{D}|$ in $\bar{\mathcal{G}}$.

If we can find $|\mathcal{D}|$ edge-independent paths from o to $\bar{d}$ in $\bar{\mathcal{G}}$, then a simple transformation that removes the last edge creates the desired set $\mathcal{P}$ in $\mathcal{G}$. On the other hand, the construction of a maximum number of independent paths in a graph between two nodes is a classical graph theory combinatorial problem for which many possible algorithms can be used. An efficient and popular approach is to transform the problem into a network flow problem and then run the Ford and Fulkerson's (1956) algorithm on it. We follow this approach not only to solve Problem 1 but also to address Problems 2 and 3. Below we briefly recall the basic notation for network flows.

3.2 1-Legibility and Network Flows

A *network* is a four tuple $\mathcal{N} = \langle \mathcal{G}, s, t, u \rangle$ where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed graph, $s \in \mathcal{V}$ is the source vertex, $t \in \mathcal{V}$ is the sink vertex and $u : \mathcal{E} \to \mathbb{R}^+$ is a positive *capacity* function that assigns a capacity $u(e)$ to each edge $e \in \mathcal{E}$. A *flow* f on a network $\mathcal{N}$ is a function $f : \mathcal{E} \to \mathbb{R}^+$ that satisfies the following two conditions: i) $\forall e \in \mathcal{E}, 0 \leq f(e) \leq u(e)$ (edge capacity constraint); and ii) $\forall v \in \mathcal{V} \setminus \{s, t\}, \sum_{e \,\in\, out(v)} f(e) = \sum_{e \,\in\, in(v)} f(e)$ (conservation of flow constraint). The *value* of a flow f is defined as the total amount of flow that leaves s: $x(f) = \sum_{e \in out(s)} f(e)$. Given the conservation of flow, this is also equal to the flow that enters t, $x(f) = \sum_{e \in in(t)} f(e)$. Given a network $\mathcal{N}$, the *maximum flow problem* consists in computing a flow f^* with the maximum possible value.

The concept of maximum flow in a network is linked to the concept of minimal cut by the Max-Flow/Min-Cut theorem (Ford and Fulkerson 1956). For this, we need to use a more general concept of cut capacity: given an $s - t$-cut (A, B), the cut capacity, $c(A, B)$, is the sum of the capacities of all the edges going from set A to set B. The Max-Flow/Min-Cut theorem states that the minimum cut capacity in a network $\lambda^*(\mathcal{N})$ is the same as the maximum flow value. Efficient algorithms can be run on $\mathcal{N}$ to find the maximum flow. One of these is the above-mentioned Ford-Fulkerson's algorithm, which presents an important feature: if the capacity of every edge is integer, then the algorithm constructs a maximum flow f^* that is also integer-valued on every edge.

Given a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$, we apply the Ford-Fulkerson's algorithm to the network $\mathcal{N} = \langle \bar{\mathcal{G}}, o, \bar{d}, u \rangle$, where $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ is as defined above and $u(e) = 1$ for every $e \in \bar{\mathcal{E}}$. The output of this algorithm is a flow f^* of maximum value. If $x(f^*) = \lambda^*(\mathcal{N}) < |\mathcal{D}|$, it follows from Corollary 1 that $k(\mathcal{G}, o, \mathcal{D}) > 1$. We will see in the next section how to handle this case. On the other hand, if $x(f^*) = \lambda^*(\mathcal{N}) = |\mathcal{D}|$, we construct $|\mathcal{D}|$ independent paths in $\bar{\mathcal{G}}$ from o to $\bar{d}$ based on f^*. We do this by exploiting the fact that, since f^* is integral and $u(e) \in \{0, 1\}$, $f^*(e) \in \{0, 1\}$ for every edge e. Starting from o and using the edges e for which $f^*(e) = 1$, standard recursive techniques allow us to construct the desired paths. Finally, by dropping the last edge in such paths, we construct an $(o, \mathcal{D})$-connecting 1-legible sets of paths $\mathcal{P}$ in $\mathcal{G}$.

Problem 2 can also be approached using network flow techniques. If we have a weight cost vector W on graph $\mathcal{G}$, we extend it to a weight cost vector $\bar{W}$ on $\bar{\mathcal{G}}$ defined by $\bar{W}_e = W_e$ if $e \in \mathcal{E}$ and $\bar{W}_e = 0$ otherwise (i.e. we give zero cost to the edges connecting the destinations $\mathcal{D}$ to the fictitious destination $\bar{d}$). For a flow f on the corresponding network $\mathcal{N} = \langle \bar{\mathcal{G}}, o, \bar{d}, u \rangle$, we can define a *cost* as follows:

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

For a desired flow value x, the *minimum cost flow problem* consists of computing a flow f with value x (i.e. pushing x units of flow from s to t, subject to the conservation and capacity constraints) that minimizes the overall cost, i.e.

$$c_{\mathcal{N}}^{min}(x) = \min_{f \text{ flow } |\, x(f) = x} c(f) \qquad (2)$$

We conventionally put $c_{\mathcal{N}}^{min}(x) = \infty$ if there is no feasible flow of value x. As the maximum flow problem seen above, the minimum-cost flow problem satisfies an analogous integral property (Ahuja, Magnanti, and Orlin 1988): if the edge capacities are all integers (as it happens in our case), there exists a flow that solves Eq. (2) and is also integer-valued on every edge. This fact, together with the connection we have established between legibility and edge-independent paths, allows us to tackle Problem 2 in the context of a minimum-cost flow problem. We start analyzing the case when the legibility delay is $k = 1$ by presenting the following result.

Proposition 2 *Given a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ with weight cost vector W, consider the network $\mathcal{N} = \langle \bar{\mathcal{G}}, o, \bar{d}, u\rangle$ with extended weight cost vector $\bar{W}$. A 1-legible $(o, \mathcal{D})$-connecting set of paths $\mathcal{P}$ in $\mathcal{G}$ has minimal cost*

$$C^{min}_{(\mathcal{G},o,\mathcal{D})}(1) = c^{min}_{\mathcal{N}}(|\mathcal{D}|) \quad (3)$$

Proof We put $C^{min}(1) = C^{min}_{(\mathcal{G},o,\mathcal{D})}(1)$ and $c^{min}(|\mathcal{D}|) = c^{min}_{\mathcal{N}}(|\mathcal{D}|)$. If a 1-legible $(o, \mathcal{D})$-connecting set of walks $\mathcal{P}$ does not exist, $C^{min}(1) = \infty$ and, by Corollary 1, a flow with value $|\mathcal{D}|$ does not exist, making the cost $c^{min}_{\mathcal{N}}(|\mathcal{D}|) = \infty$. Hence, the equality is satisfied in this case. Let us now consider the case in which a 1-legible $(o, \mathcal{D})$-connecting set of paths exist and let $\mathcal{P}$ be one of minimum cost so that $C(\mathcal{P}) = C^{min}(1)$. The paths in $\mathcal{P}$ can be extended to $|\mathcal{D}|$ edge-independent paths in $\bar{\mathcal{G}}$ connecting o to $\bar{d}$ and having the same cost. Such paths define a flow f on $\mathcal{N}$ of value $|\mathcal{D}|$ that takes as possible values only 0 or 1 and, given an edge e, $f_e = 1$ if and only if e appears in one of the extended paths. By construction, this flow f has the same cost $C^{min}(1)$. This proves that $\geq$ holds in (3). Consider now an integer valued flow f of minimum cost for $\mathcal{N}$ and value $|\mathcal{D}|$. We thus have $c(f) = c^{min}_{\mathcal{N}}(1)$. The flow f only takes values 0 or 1 and the edges on which the flow is 1 form, as discussed above, $|\mathcal{D}|$ edge-independent walks in $\bar{\mathcal{G}}$ from o to $\bar{d}$. By removing the last edge, we obtain a 1-legible $(o, \mathcal{D})$-connecting set of paths $\mathcal{P}$ in $\mathcal{G}$ whose cost is $C(\mathcal{P}) = c^{min}_{\mathcal{N}}(1)$. This yields $\leq$ in expression (3) and completes the proof of the theorem. ■

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

4 Reformulating s-Legibility as 1-Legibility

Checking s-legibility can be reduced to checking 1-legibility when we operate a suitable transformation of the original graph $\mathcal{G}$. Given a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ and $s \in \mathbb{N}$, with $s > 1$, we construct a new graph $\mathcal{G}^{(s)}$, which we call the *s-legibility graph*, 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. This construction ensures that, if we find two edge-independent walks in the s-legibility graph, they correspond to walks in the original graphs that can share $(s-1)$ elements at the most. This, in turn, means that, by checking 1-legibility in the s-legibility graph, we can draw conclusions on the s-legibility in the original graph. To make these concepts precise, we introduce a number of auxiliary sets.

4.1 s-Legibility Graph's Construction

In this section, we explicitly represent $\mathcal{D} = \{d_1, \ldots, d_q\}$. We then start by taking all walks in $\mathcal{G}$ of a given length s:

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

Given $\gamma = (e_1, \ldots, e_s) \in \mathcal{Q}^{(s)}$, we put $\gamma_<$ for $(e_1, \ldots, e_{s-1})$ and $\gamma_>$ for $(e_2, \ldots, e_s)$.

We then consider the following subsets of $\mathcal{Q}^{(s)}$, respectively, (i) walks that start in o; (ii) walks that end in a specific destination d_i; (iii) walks that end in any destination in $\mathcal{D}$; and (iv) walks that do not start in o and end in $\mathcal{D}$:

$$\begin{array}{lll}
\text{(i)} & \mathcal{Q}^{(s)}_{(o} & = \{(e_1, \ldots, e_s) \in \mathcal{Q}^s \mid \theta(e_1) = o\}, \\
\text{(ii)} & \mathcal{Q}^{(s)}_{d_i)} & = \{(e_1, \ldots, e_s) \in \mathcal{Q}^{(s)} \mid \kappa(e_s) = d_i\}, \\
\text{(iii)} & \mathcal{Q}^{(s)}_{\mathcal{D})} & = \bigcup_{d_i \in \mathcal{D}} \mathcal{Q}^{(s)}_{d_i)}, \\
\text{(iv)} & \mathcal{Q}^{(s)}_{\bullet} & = \mathcal{Q}^{(s)} \setminus \left(\mathcal{Q}^{(s)}_{(o} \cup \mathcal{Q}^{(s)}_{\mathcal{D})}\right)
\end{array}$$

We introduce a virtual origin $\bar{o}$, a set of virtual destinations that is in one-to-one correspondence with the original set of destinations $\mathcal{D}$, i.e. $\bar{\mathcal{D}} = \{\bar{d}_1, \ldots, \bar{d}_q\}$, and the following subset of $\bar{\mathcal{D}}$ that corresponds to destinations reachable in a number of steps smaller than s:

$$\bar{\mathcal{D}}^{(s)} = \{\bar{d}_i \mid \exists\, \gamma \text{ walk in } \mathcal{G} \text{ from } o \text{ to } d_i,\ l_\gamma \leq s-1\} \quad (4)$$

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

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

with $\mathcal{E}^{(s)}_{\bar{o}-\bar{\mathcal{D}}} = \{e_{\bar{d}_i} \mid \bar{d}_i \in \bar{\mathcal{D}}^{(s)}\}$, a set containing edges in one-to-one correspondence with the virtual destinations in $\bar{\mathcal{D}}^{(s)}$. Tail and head functions are respectively $\theta : \mathcal{E}^{(s)} \to \mathcal{V}^{(s)}$ and $\kappa : \mathcal{E}^{(s)} \to \mathcal{V}^{(s)}$. They are defined based on whether $e \in \mathcal{E}^{(s)}$ belongs to $\mathcal{Q}^{(s)}_{\bullet}$, $\mathcal{Q}^{(s)}_{(o}$, $\mathcal{Q}^{(s)}_{\mathcal{D})}$, or $\mathcal{E}^{(s)}_{\bar{o}-\bar{D}}$, as follows:

$$\begin{array}{lll}
\gamma \in \mathcal{Q}^{(s)}_{\bullet} & \theta(\gamma) = \gamma_<, & \kappa(\gamma) = \gamma_> \\
\gamma \in \mathcal{Q}^{(s)}_{(o} & \theta(\gamma) = \bar{o}, & \kappa(\gamma) = \gamma_> \\
\gamma \in \mathcal{Q}^{(s)}_{d_i)} & \theta(\gamma) = \gamma_<, & \kappa(\gamma) = \bar{d}_i \\
e_{\bar{d}_i} \in \mathcal{E}^{(s)}_{\bar{o}-\bar{\mathcal{D}}} & \theta(e) = \bar{o}, & \kappa(e) = \bar{d}_i
\end{array}$$

Let us now see the interpretation of the graph. 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 $s-1$, namely $\gamma_<$ 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 two exceptions: any walk γ that starts in o is instead considered as an edge starting from the virtual origin $\bar{o}$ and ending in $\gamma_>$; and a walk that ends in a destination d_i is instead considered as an edge ending in the corresponding virtual destination $\bar{d}_i$ while starting in $\gamma_<$. Also, for each destination d_i for which, in the original graph, there is a walk of length smaller than s connecting the origin o to d_i, there is an edge connecting the virtual origin $\bar{o}$ to the corresponding virtual destination $\bar{d}_i$ in the s-legibility graph. These edges are necessary to represent walks that would not otherwise appear in the s-legibility graph.

If the original graph $\mathcal{G}$ is equipped with a weight vector $W \in \mathbb{R}^{\mathcal{E}}_+$, this is extended to a weight $W^{(s)}$ in the

s-legibility graph $\mathcal{G}^{(s)}$ in the following way. For edges $(e_1, \ldots, e_s) \in \mathcal{Q}^{(s)}$, we put

$$(e_1, \ldots, e_s) \in \mathcal{Q}^{(s)} \setminus \mathcal{Q}^{(s)}_{\mathcal{D})} \;\rightarrow\; W^{(s)}_{(e_1,\ldots,e_s)} = W_{e_1}$$
$$(e_1, \ldots, e_s) \in \mathcal{Q}^{(s)}_{\mathcal{D})} \;\rightarrow\; W^{(s)}_{(e_1,\ldots,e_s)} = W_{e_1} + \cdots + W_{e_s} \tag{6}$$

Finally, for each of the edges $e_{\bar{d}_i} \in \mathcal{E}^{(s)}_{\bar{o}-\bar{\mathcal{D}}}$, we put $W^{(s)}_{e_{\bar{d}_i}}$ equal to the minimum cost among all possible walks from o to d_i in $\mathcal{G}$ having length $< s$. Costs for walks and set of walks will be denoted with the symbol $C^{(s)}$. In tackling Problems 2 and 3, we will assume that the weight vectors on the s-legibility graphs are defined in the way just described.

4.2 From s-Legibility to 1-Legibility

We now 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 $\mathcal{G}^{(s)}$. This is achieved via the following natural transformations between walks in $\mathcal{G}$ and $\mathcal{G}^{(s)}$:

- Given a walk $\gamma = (e_1, \ldots, e_l)$ in $\mathcal{G}$ of length $l \geq s$ connecting o to some destination d_i, we can consider the corresponding walk γ' in $\mathcal{G}^{(s)}$ given by:
  $$\gamma' = \Gamma^f(\gamma) = ((e_1, \ldots, e_s), \ldots, (e_{l-s+1}, \ldots, e_l)) \tag{7}$$
  Note that, by the definition of the tail and head functions for $\mathcal{G}^{(s)}$, we have that γ' connects $\bar{o}$ to $\bar{d}_i$.
- Conversely, we notice that any walk γ' from $\bar{o}$ to $\bar{d}_i$ in the subgraph $\tilde{\mathcal{G}}^{(s)} = (\mathcal{V}^{(s)}, \mathcal{Q}^{(s)})$ of $\mathcal{G}^{(s)}$ (i.e. not using edges in $\mathcal{E}^{(s)}_{\bar{o}-\bar{\mathcal{D}}}$) must necessarily be of the form (7). For such a walk, we define
  $$\gamma = \Gamma^b(\gamma') = (e_1, \ldots, e_l)$$

We notice that, by construction,

$$\Gamma^b\Gamma^f(\gamma) = \gamma, \quad \Gamma^f\Gamma^b(\gamma') = \gamma'$$

Proposition 3 *The following facts hold true:*

1. *Consider a set $\mathcal{P}$ of walks in $\mathcal{G}$ from o to $\mathcal{D}$, all of length $l \geq s$, that is $(o, \mathcal{D})$-connecting and s-legible. Then, $\Gamma^f(\mathcal{P})$ is $(\bar{o}, \bar{\mathcal{D}})$-connecting and 1-legible. Moreover, $C^{(s)}(\Gamma^f(\mathcal{P})) = C(\mathcal{P})$.*
2. *Consider a set $\mathcal{P}'$ of walks in $\tilde{\mathcal{G}}^{(s)}$ from $\bar{o}$ to $\bar{\mathcal{D}}$ that is $(\bar{o}, \bar{\mathcal{D}})$-connecting and 1-legible. Then, $\Gamma^b(\mathcal{P}')$ is $(o, \mathcal{D})$-connecting and s-legible. Also, $C(\Gamma^b(\mathcal{P}')) = C^{(s)}(\mathcal{P}')$.*

Proof 1. Consider the walk $\gamma = (e_1, \ldots, e_l) \in \mathcal{P}$ that connects o to d_i. Then, $\Gamma^f(\gamma)$ connects $\bar{o}$ to $\bar{d}_i$. Since $\mathcal{P}$ is $(o, \mathcal{D})$-connecting, this implies that $\Gamma^f(\mathcal{P})$ is $(\bar{o}, \bar{\mathcal{D}})$-connecting. Considering that, by construction, the edges appearing in the walks in $\Gamma^f(\mathcal{P})$ correspond to the subwalks of length s of the walks in $\mathcal{P}$, the s-legibility of $\mathcal{P}$ yields the 1-legibility of $\Gamma^f(\mathcal{P})$. Finally, the cost equality follows from the fact that, because of relations (6), $C^{(s)}(\Gamma^f(\gamma)) = C(\gamma)$.

2. It follows from arguments analogous to 1, after we notice the following two facts. First, if $\gamma' \in \mathcal{P}'$ connects $\bar{o}$ to $\bar{d}_i$, then $\Gamma^b(\gamma')$ connects o to d_i. Second, all possible subwalks of length s in $\Gamma^b(\gamma')$ are edges in γ'. ■

The following relationship between legibility in $\mathcal{G}$ and $\mathcal{G}^{(s)}$ holds.

Theorem 1 *Consider a legibility problem instance $(\mathcal{G}, o, \mathcal{D})$ and, given $s > 1$, the auxiliary legibility problem instance $(\mathcal{G}^{(s)}, \bar{o}, \bar{\mathcal{D}})$. Then $(\mathcal{G}, o, \mathcal{D})$ is s-legible if and only if $(\mathcal{G}^{(s)}, \bar{o}, \bar{\mathcal{D}})$ is 1-legible. Moreover,*

$$C^{min}_{(\mathcal{G},o,\mathcal{D})}(k) = C^{min}_{(\mathcal{G}^{(s)},\bar{o},\bar{\mathcal{D}})}(1) \tag{8}$$

Proof 'If': Suppose $\bar{\mathcal{P}}$ is a 1-legible $(\bar{o}, \bar{\mathcal{D}})$-connecting set of walks in $\mathcal{G}^{(s)}$. We partition $\bar{\mathcal{P}}$ into two parts $\bar{\mathcal{P}} = \bar{\mathcal{P}}_\bullet \cup \bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}$, where $\bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}$ contains all the walks of length 1 in $\bar{\mathcal{P}}$ that consist of an edge in $\mathcal{E}^{(s)}_{\bar{o}-\bar{\mathcal{D}}}$. We call $\mathcal{D}_\bullet$ and $\mathcal{D}_{<s}$ the set of those destinations $d_i \in \mathcal{D}$ for which the virtual corresponding one $\bar{d}_i$ is reached by a walk, respectively, in $\bar{\mathcal{P}}_\bullet$ or in $\bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}$. By construction $\mathcal{D}_\bullet \cup \mathcal{D}_{<s} = \mathcal{D}$. For every walk $e_{\bar{d}_i} \in \bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}$ (where, we recall, $\bar{d}_i \in \bar{\mathcal{D}}^{(s)}$) we now consider a walk $\gamma^{(d_i)}$ in $\mathcal{G}$ from o to d_i of length $l \leq s-1$ and minimum cost. This must exist given the definition of $\bar{\mathcal{D}}^{(s)}$. We put

$$\mathcal{P} = \Gamma^b(\bar{\mathcal{P}}_\bullet) \cup \{\gamma^{(d_i)} \mid e_{\bar{d}_i} \in \bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}\} \tag{9}$$

By Proposition 3, $\Gamma^b(\bar{\mathcal{P}}_\bullet)$ is $(o, \mathcal{D}_\bullet)$-connecting and s-legible, while $\{\gamma^{(d_i)} \mid e_{\bar{d}_i} \in \bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}\}$ is $(o, \mathcal{D}_{<s})$-connecting. Since this last set only consists of walks of length below s, Remark 1 allows to conclude that $\mathcal{P}$ is $(o, \mathcal{D})$-connecting and s-legible.

It also follows from Proposition 3 that $C(\Gamma^b(\bar{\mathcal{P}}_\bullet)) = C^{(s)}(\bar{\mathcal{P}}_\bullet)$. Given the definition of weight for the directed edges in $\mathcal{E}^{(s)}_{\bar{o}-\bar{\mathcal{D}}}$ and the optimal way in which we choose the walks $\gamma^{(d_i)}$, we also have that $C(\gamma^{(d_i)}) = C^{(s)}(\bar{e}_{d_i})$ for every $d_i \in \mathcal{D}_{<s}$. Thus, $C(\{\gamma^{(d_i)} \mid e_{\bar{d}_i} \in \bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}}\}) = C^{(s)}(\bar{\mathcal{P}}_{\bar{o}-\bar{\mathcal{D}}})$. As the two sets are disjoint, this shows that $C(\mathcal{P}) = C^{(s)}(\bar{\mathcal{P}})$ and this proves $\leq$ in relation (8).

'Only if': Suppose $\mathcal{P}$ is an s-legible $(o, \mathcal{D})$-connecting set of walks in $\mathcal{G}$. Write $\mathcal{P} = \mathcal{P}_\bullet \cup \mathcal{P}_{<s}$, where $\mathcal{P}_{<s}$ consists of the walks of length smaller than s and $\mathcal{P}_\bullet$ consists of the remaining ones. Let $\mathcal{D}_{<s}$ and $\mathcal{D}_\bullet$ be the set of destinations reached by the walks, respectively, in $\mathcal{P}_{<s}$ and in $\mathcal{P}_\bullet$. We indicate the corresponding subsets in $\bar{\mathcal{D}}$ as, respectively, $\bar{\mathcal{D}}_{<s}$ and $\bar{\mathcal{D}}_\bullet$. By point 1 of Proposition 3, $\Gamma^f(\mathcal{P}_\bullet)$ is a 1-legible $(\bar{o}, \bar{\mathcal{D}}_\bullet)$-connecting set of walks in $\mathcal{G}^{(s)}$. On the other hand, the set

$$\bar{\mathcal{P}}_{<s} = \{e_{\bar{d}_i} \mid d_i \in \mathcal{D}_{<s}\} \tag{10}$$

is a 1-legible $(\bar{o}, \bar{\mathcal{D}}_{<s})$-connecting set of walks in $\mathcal{G}^{(s)}$. Since edges used in $\Gamma^f(\mathcal{P}_\bullet)$ and $\bar{\mathcal{P}}_{\leq s}$ form disjoint sets and, by construction, $\bar{\mathcal{D}} = \bar{\mathcal{D}}_{<s} \cup \bar{\mathcal{D}}_\bullet$, we conclude that $\bar{\mathcal{P}} = \Gamma^f(\mathcal{P}_\bullet) \cup \bar{\mathcal{P}}_{<s}$ is a 1-legible $(\bar{o}, \bar{\mathcal{D}})$-connecting set of walks in $\mathcal{G}^{(s)}$.

Following the same line of reasoning as above, we have that $C^{(s)}(\Gamma^f(\mathcal{P}_\bullet)) = C(\mathcal{P}_\bullet)$ thanks to Proposition 3 and $C^{(s)}(\bar{\mathcal{P}}_{<s}) \leq C(\mathcal{P}_{<s})$ by the way $\bar{\mathcal{P}}_{<s}$ has been defined in (10). Thus, $C^{(s)}(\bar{\mathcal{P}}) \leq C(\mathcal{P})$. This proves $\geq$ in relation (8) and completes the proof. ■

Example 2 *In Figure 2, we present a fragment of graph $\mathcal{G}^{(2)}$ corresponding to graph $\mathcal{G}$ in Figure 1(a). Edges are*

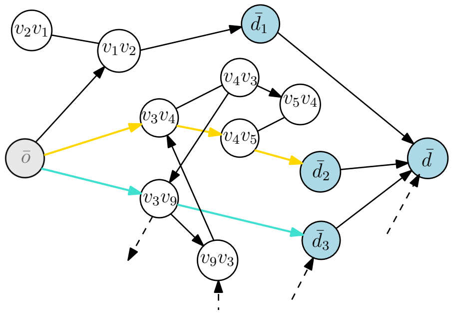
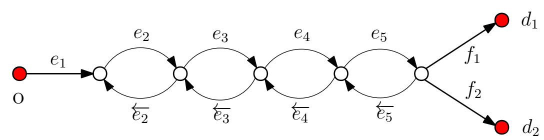


Figure 2: A fragment of graph $\mathcal{G}^{(2)}$ for graph $\mathcal{G}$ in Fig. 1(a).

denoted as ordered pair of vertices (as no parallel edge is present in the original graph). The fictitious destination is $\bar{d}$. The two colored walks correspond to the two walks in Figure 1(b) leading to d_2 and d_3. Notice how, while the two original walks in $\mathcal{G}$ in Figure 1(b) have a common edge, the two walks in $\mathcal{G}^{(2)}$ are instead edge-independent.

An interesting final remark is that we cannot extend Proposition 1 to legibility delay higher than 1. Namely, it is possible that, for certain problems, there are s-legible sets of walks $\mathcal{P}$ but not s-legible sets of paths. This is shown in the example below for a simple case with only two destinations.

Example 3 *Consider the graph in the picture below.*



Any path from o to the destinations will share the subpath $\gamma = (e_1e_2e_3e_4e_5)$. Therefore, every $(o, \mathcal{D})$-connecting set of paths $\mathcal{P}$ will necessarily exhibits legibility delay $k(\mathcal{P}) = 6$. On the other hand, we can consider the following two walks:

$$\begin{aligned} \gamma_1 &= e_1e_2e_3e_4e_5f_1 \\ \gamma_2 &= e_1e_2\overleftarrow{e_2}e_2e_3\overleftarrow{e_3}e_3e_4\overleftarrow{e_4}e_4e_5\overleftarrow{e_5}e_5f_2 \end{aligned} \tag{11}$$

$\tilde{\mathcal{P}} = \{\gamma_1, \gamma_2\}$ *is $(o, \mathcal{D})$-connecting and $k(\tilde{\mathcal{P}}) = 3$.*

The idea behind Example 3 is that coding strategies can be used to lower the legibility delay. If we allow the motion of the agents to contain closed loops, these can be exploited to significantly reduce the legibility delay. A budget constraint would automatically bound the use of this feature.

5 Algorithms

Given a graph $\mathcal{G}$, Theorem 1 allows us to tackle Problem 1 by studying the 1-legibility of the s-legibility graphs $\mathcal{G}^{(s)}$. If we now combine this result with the considerations in Section 3, we can use network flow algorithms to solve all these 1-legibility problems. Algorithms 1 and 2 (see below) for computing legibility are directly inspired by Theorem 1. Algorithm 1 takes a legibility instance $(\mathcal{G}, o, \mathcal{D})$ and returns its legibility delay and a set of paths $\mathcal{P}$ such that $k(\mathcal{P}) = k(\mathcal{G}, o, \mathcal{D})$. The function $createLegibilityGraph()$ at line 8 calls Algorithm 2, which constructs the s-legibility graph $\mathcal{G}^{(s)}$ based on the s-legibility graph $\mathcal{G}^{(s-1)}$ at the previous iteration. The function $extractDestinations()$ at line 10 implements expression (4) and $constructGraph()$ at line 12 implements expression (5). The complexity of the algorithms is dominated by the double loop at lines 1 and 2 of Algorithm 2, leading to a complexity of $\mathcal{O}(|\mathcal{E}^{(s-1)}|^2)$.

Algorithm 1: Legibility Delay

Data: $\Pi = (\mathcal{G}, o, \mathcal{D})$
Result: $k(\mathcal{G}, o, \mathcal{D})$
$\qquad \mathcal{P} \mid k(\mathcal{P}) = k(\mathcal{G}, o, \mathcal{D})$
1 $\mathcal{N} \leftarrow createNetwork(\bar{\mathcal{G}})$
2 $\mathcal{P}, \lambda^*(\bar{\mathcal{G}}, o, \bar{d}) \leftarrow FordFulkerson(\mathcal{N})$
3 **if** $\lambda^*(\bar{\mathcal{G}}, o, \bar{d}) < |\mathcal{D}|$ **then**
4 $\quad s \leftarrow 1$
5 $\quad \mathcal{G}^{(1)} \leftarrow \bar{\mathcal{G}}$
6 $\quad$ **do**
7 $\quad\quad s \leftarrow s + 1$
8 $\quad\quad \mathcal{G}^{(s)} \leftarrow createLegibilityGraph(\mathcal{G}^{(s-1)})$
9 $\quad\quad \mathcal{N}^{(s)} \leftarrow createNetwork(\mathcal{G}^{(s)})$
10 $\quad\quad \mathcal{P}, \lambda^*(\mathcal{G}^{(s)}, o, \bar{d}) \leftarrow FordFulkerson(\mathcal{N}^{(s)})$
11 $\quad$ **while** $\lambda^*(\mathcal{G}^{(s)}, o, \bar{d}) < |\mathcal{D}|$
12 $k(\mathcal{G}, o, \mathcal{D}) = k(\mathcal{P})$

Algorithm 2: Construction of s-legibility graph $\mathcal{G}^{(s)}$

Data: $\mathcal{G}^{(s-1)}$
Result: $\mathcal{G}^{(s)}$
1 **foreach** $(e_1, \dots, e_{s-1}) \in \mathcal{Q}^{(s-1)}$ **do**
2 $\quad$ **foreach** $(e'_1, \dots, e'_{s-1}) \in \mathcal{Q}^{(s-1)}$ **do**
3 $\quad\quad$ *add = true*
4 $\quad\quad$ **foreach** $1 \leq k \leq (s-2)$ **do**
5 $\quad\quad\quad$ **if** $e'_k \neq e_{k+1}$ **then**
6 $\quad\quad\quad\quad$ *add = false*
7 $\quad\quad\quad\quad$ **break**
8 $\quad\quad$ **if** *add = true* **then**
9 $\quad\quad\quad Q^{(s)} \leftarrow (e_1, \dots, e_{s-1}, e'_{s-1})$
10 $\bar{\mathcal{D}}^{(s-1)} \leftarrow extractDestinations(\mathcal{G}^{(s-1)})$
11 $\bar{\mathcal{D}}^{(s)} \leftarrow \bar{\mathcal{D}}^{(s-1)} \cup \{\bar{d}_i \in \bar{\mathcal{D}} \mid \exists\, (e_1, \dots, e_{s-1}) \in \mathcal{Q}^{(s-1)}_{(o} \mid \kappa(e_{s-1}) = d_i)$
12 $\mathcal{G}^{(s)} \leftarrow constructGraph(\bar{\mathcal{D}}^{(s-1)}, \mathcal{G}^{(s)})$

Following Proposition 2 and subsequent considerations in Section 3, we solve Problem 2 for legibility delay 1 by applying the classical Primal-Dual algorithm (Ahuja, Magnanti, and Orlin 1988). Similarly to Problem 1, Problem 2 for a generic legibility delay s can be transformed into the corresponding optimization problem for $\mathcal{G}^{(s)}$ with legibility delay 1.

We now briefly tackle Problem 3 and propose a solution that exploits its duality with Problem 2. Given a budget B, the goal is to compute $k^{min}_{(\mathcal{G},o,\mathcal{D})}(B)$. We proceed as follows. We start from the legibility delay $k_1 = k(\mathcal{G}, o, \mathcal{D})$ and compute, using the previous algorithms, $B_1 = C^{min}_{(\mathcal{G},o,\mathcal{D})}(k_1)$. We then put $k^{min}_{(\mathcal{G},o,\mathcal{D})}(B) = k_1$ for every $B \geq B_1$. We then take the next $k_2 > k_1$ such that $B_2 = C^{min}_{(\mathcal{G},o,\mathcal{D})}(k_2) < B_1$ and put $k^{min}_{(\mathcal{G},o,\mathcal{D})}(B) = k_2$ for every $B_2 \leq B < B_1$. We iterate in such a way until the budget has reached the minimum value that is necessary to obtain an $(o,\mathcal{D})$-connecting set of walks. We obtain a step function for $k^{min}_{(\mathcal{G},o,\mathcal{D})}(B)$, where each cost interval corresponds to the smallest legibility delay k that can be found by incurring that cost.

6 Experimental Results

For the experiments, we use a cluster with Intel Xeon E5-2640 processors running at 2.60GHz. The memory limit by process is set to 4 GBs and time limit to 172,800 seconds. We use grids with random obstacles as they resemble physical structures such as building and plant floors. We create a fully connected graph and then progressively reduce its connectivity. More specifically, we build a four-way connected 2D orthogonal grid. To vary its connectivity, we randomly add obstacles using different thresholds (10%, 30%, 50%) and a different seed for each problem instance. We set the grid size to 30x30 with destinations varying from 2 to 8 with 2 unit increments. We run 150 problems per category (obstacles ratio, $|\mathcal{D}|$) and report average values. For all experiments, origin nodes are chosen randomly. Destinations are also chosen randomly among those nodes that are reachable from the origin node.

Figures 3a and 3b show results relating to Problem 1. All the results are averaged out over all problems as a function of the obstacle ratio. Figure 3a displays the average time (in seconds) it takes to calculate the legibility delay when we vary the number of destinations and the obstacle ratio. The legibility delay is harder to compute for 30% grids than for 10% ones because the average k level is lower when there are only a few obstacles. The same logic does not apply when increasing the obstacle ratio to 50%. The 50% problems are easier to solve than the 30% ones because there is a limited number of paths that can reach all destinations. Hence, even though the average k is higher for the 50% case, the ratio between path availability and minimum legibility delay leads to higher difficulty for the 30% problems. Figure 3b shows that, as expected, the legibility delay increases as we increase the number of destinations and obstacle rate.

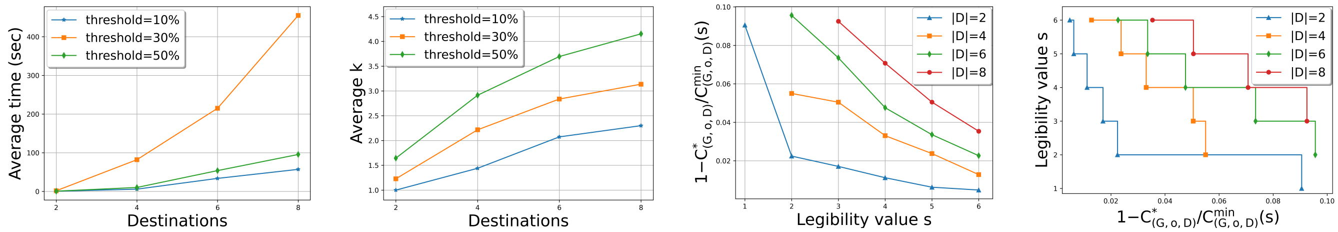
To be able to assess performance for Problems 2 and 3, we consider the index $1 - C^{*}_{(\mathcal{G},o,\mathcal{D})}/C^{min}_{(\mathcal{G},o,\mathcal{D})}(s)$, where $C^{*}_{(\mathcal{G},o,\mathcal{D})}$ is the minimum cost of an $(o,\mathcal{D})$-connecting set of walks in $\mathcal{G}$. This index can be interpreted as a distance from optimality when we constrain the legibility delay. Figure 3c focuses on Problem 2 and displays the average distance to optimality as a function of the legibility delay s for 30% obstacle rate. The figure shows that the more destinations, the higher the distance from optimality. As we accept higher legibility delays, the cost decreases. Figure 3d focuses on Problem 3, showing that, as expected, having a bigger budget allows us to decrease the legibility delay. As the number of destinations increases, the effect of the budget is less significant.

7 Related Work

In the AI and robotics communities, there has been growing interest in *interpretable agent behavior* in the past few years (Dragan, Lee, and Srinivasa 2013; Langley et al. 2017; Gunning and Aha 2019; Chakraborti et al. 2019; Sreedharan et al. 2021), 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 (Bhatt, Ravikumar, and Moura 2019). Interpretability has been studied along three main dimensions, *legibility, explicability* and *predictability* (Chakraborti et al. 2019), but, lately, some effort has been made to connect and integrate these concepts in unified frameworks (Sreedharan et al. 2021; Miura and Zilberstein 2021). We will limit our discussion to legibility and the most relevant related work.

The broad notion of legibility that we use has been proposed by Dragan et al. (2013), who introduced it in the context of motion planning. The setting of our work is, however, different from Dragan et al.'s one. 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 planner searches for a plan towards the agent's true goal by favoring actions that maximize the goal's posterior probability. On the other hand, we assume the observer is deterministic as its decisions are completely determined by its knowledge of the agents' walks and its observations of the agent's behavior. We build plans from the origin to all possible goals, of which the observer has full knowledge. Based on such knowledge, if it observes a transition (an edge) that belongs to one path only, it can decide where the agent is going with certainty; conversely, if it observes a transition that belongs to more than one path, it defers a decision until it observes the first transition that belongs to one path only.

Our technique bears some similarities with the work of Keren et al. (2019) on GRD. In particular, the notion of worst case distinctiveness (*wcd*), which is used in GRD, is connected with our legibility delay. 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



(a) Runtime for calculating the legibility delay.

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

(c) Change in budget to reach different legibility delays.

(d) Change in the legibility delay determined by different budgets.

Figure 3: Experiments on the legibility delay for a 30×30 grid with destinations varying from 2 to 8 and 10%, 30%, 50% obstacle ratios.

not alter the environment; we construct legible paths in it.

We define legibility in the context of path finding, which, given its importance in several practical applications, have been studied for many years. We refer the readers to recent survey papers on this topic for an overview of them (Stern 2019). Here, we focus on the connection between our work and Yu and La Valle's algorithm for solving anonymous multi-agent path finding problems (Yu and LaValle 2013). 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.

8 Conclusions and Future Work

We explore the concept of goal legibility in a path planning scenario. We present three optimization problems through which we study how to achieve minimum legibility and how minimum legibility is influenced by the agent's available budget. We show that those problems can be transformed into classical optimization network flow problems when we make suitable transformations on the graph underlying the path planning problem. 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 and to study more sophisticated settings, e.g. partially observable environments. In line with our goal of simplifying the decision-making process of the observer and reducing their cognitive load, we plan to study the most natural way to present the observer with an association between subwalks and goals as the look-up tables mentioned in Section 2 will not be suitable in complex environments. We will run human studies to make this assessment.

Acknowledgements

We thank the anonymous reviewers for their useful comments. This work has been conducted with the support of EPSRC (Grant EP/S016473/1), Leverhulme Trust (Grant VP1-2019-037), and MIUR (Grant CUP: E11G18000350001).

Ethics Statement

Our technique is motivated by autonomous robotic missions in challenging scenarios, where the agents and the humans do not share the same physical space and do not interact directly. Instead, the robots autonomously perform some tasks, and a human supervisor remotely oversees 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 (direct communication might not be available). It follows that we envisage a positive use of our technique to simplify the decision-making process of the human operators and reduce their cognitive load. However, considering potential misuses of our technique, any method that facilitates legibility could potentially be turned into a surveillance system. In our case, if the agents are provided with the walks to follow without informing them that such walks maximize legibility, their behavior could be easily surveilled by a malicious actor. Open-access research usually discourages malicious use of technology as the techniques are in the public domain after being published.

References

Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1988. *Network flows*. Cambridge, Mass.: Alfred P. Sloan School of Management, Massachusetts.

Bhatt, U.; Ravikumar, P.; and Moura, J. M. F. 2019. Building Human-Machine Trust via Interpretability. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 9919–9920.

Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D. E.; and Kambhampati, S. 2019. Explicability? Legibility? Predictability? Transparency? Privacy? Security? The Emerging Landscape of Interpretable Agent Behavior. In *Proc. of the Twenty-Ninth International Conference on Automated*

Planning and Scheduling, ICAPS 2019, Berkeley, CA, USA, 86–96.

Dragan, A.; and Srinivasa, S. 2014. Integrating Human Observer Inferences into Robot Motion Planning. *Auton. Robots*, 37(4): 351–368.

Dragan, A. D. 2017. Robot Planning with Mathematical Models of Human State and Action. *CoRR*, abs/1705.04226.

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

Ford, L. R.; and Fulkerson, D. R. 1956. Maximal Flow Through a Network. *Canadian Journal of Mathematics*, 8: 399–404.

Gunning, D.; and Aha, D. 2019. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine*, 40(2): 44–58.

Keren, S.; Gal, A.; and Karpas, E. 2019. Goal Recognition Design in Deterministic Environments. *Journal of Artificial Intelligence Research (JAIR)*, 65.

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

Kulkarni, A.; Srivastava, S.; and Kambhampati, S. 2020. 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*, AAMAS '20, 1889–1891. International Foundation for Autonomous Agents and Multiagent Systems.

Langley, P.; Meadows, B.; Sridharan, M.; and Choi, D. 2017. Explainable Agency for Intelligent Autonomous Systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 4762–4763. AAAI Press.

MacNally, A. M.; Lipovetzky, N.; Ramirez, M.; and Pearce, A. R. 2018. Action Selection for Transparent Planning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, 1327–1335. Richland, SC: International Foundation for Autonomous Agents and Multi-Agent Systems.

Menger, K. 1927. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10(1): 96–115.

Miura, S.; Cohen, A. L.; and Zilberstein, S. 2021. Maximizing Legibility in Stochastic Environments. In *2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, 1053–1059.

Miura, S.; and Zilberstein, S. 2021. A Unifying Framework for Observer-Aware Planning and its Complexity. In *37th Conference on Uncertainty in Artificial Intelligence (UAI 2021)*.

Sreedharan, S.; Kulkarni, A.; Smith, D. E.; and Kambhampati, S. 2021. 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)*.

Stern, R. 2019. *Multi-Agent Path Finding – An Overview*, 96–115. Springer International Publishing.

Yu, J.; and LaValle, S. M. 2013. Multi-agent Path Planning and Network Flow. In Frazzoli, E.; Lozano-Perez, T.; Roy, N.; and Rus, D., eds., *Algorithmic Foundations of Robotics X*, 157–173. Springer Berlin Heidelberg.