Economical Graph Discovery *

Noga Alon
Tel-Aviv University and Microsoft Research Israel, nogaa@post.tau.ac.il

Yuval Emek
ETH, Zurich, yuval.emek@tik.ee.ethz.ch

Michal Feldman
Hebrew University of Jerusalem and Harvard University, mfeldman@seas.harvard.edu

Moshe Tennenholtz
Microsoft Research Israel and Technion-Israel Institute of Technology, moshet@microsoft.com

We consider the problem of *graph discovery* in settings where the graph topology is known, while the edge weights are hidden. The setting consists of a weighted graph G with n vertices and m edges, and with designated source s and destination t. We study two different *discovery* problems; namely, (i) edge weight discovery, where the goal is to discover all edge weights, and (ii) shortest-path discovery, where the goal is to discover a shortest (s, t)-path. This discovery is done by means of *agents* that traverse different (s, t)-paths in multiple *rounds* and report back the total cost they incurred. Three *cost models* are considered, differing from each other in their approach to congestion effects. In particular, we consider congestion-free models as well as models with positive and negative congestion effects. We seek bounds on the number of rounds and the number of agents required to complete the edge-weights or the shortest-path discovery.

A number of results concerning such bounds for both directed and undirected graphs are established. Among these results, we show that: (1) for undirected graphs, all edge weights can be discovered within a single round consisting of m agents; (2) discovering a shortest path in either undirected or directed acyclic graphs requires at least $m - n + 1$ agents; and (3) the edge weights in a directed acyclic graph can be discovered in m rounds with $m + n - 2$ agents under congestion-aware cost models. Our study introduces a new setting of graph discovery under uncertainty and provides fundamental understanding of the problem.

Key words: graph discovery, shortest path, congestion, cost sharing

1. Introduction

Suppose you are in an unfamiliar environment and wish to arrive at some designated destination. You can use a map to learn the different routes toward the destination, but without information on the actual delays on the different roads, it will be difficult to determine how long each route will take. To obtain more accurate information about the delays along the different roads, or merely to identify the shortest way toward the destination, you may consult with people who have already taken particular routes. While people rarely remember the duration of each portion along their

* A preliminary version of this paper appeared in the 2nd International Conference on Innovations in Computer Science (ICS), Beijing, 2011.

routes, they can often report back the total time duration of their trips. The goal then is to use this aggregated information to make inference about the individual road delays, or to identify the shortest path.

Scenarios of this sort typically arise in the context of network routing. While the underlying infrastructure is usually stable over long periods of times, the delays on the network edges can be easily modified over short time periods. Therefore, it is reasonable to assume that nodes in the network are aware of the network topology, while they are uncertain about the delays on each link. Learning the link delays may be crucial for routing decisions. This information can be gathered by transmitting packets along the different paths and observing their trip times. Since one cannot assume the participation of all the nodes in the network, end-to-end measurements have become a common tool in network measurements (see, e.g., Shih and Hero 2002). The challenge is then to use the end-to-end measurements in order to make meaningful network inferences in an efficient way.

We model situations like the above using a network discovery model. In our model, a weighted graph with known topology but unknown edge weights is given, along with designated source and destination nodes (s and t, respectively). Our goal is to devise a discovery protocol that learns the edge weights, or one that achieves the simpler task of identifying the shortest s-t path. The discovery process is operated through agents that traverse the network and report back their total cost. The process proceeds in multiple rounds, where multiple agents can be sent in each round. The objective is to devise such protocols of minimum complexity, where the complexity of a protocol is measured according to the number of rounds of transmissions, and the number of participating agents.

As we shall see, the complexity of a protocol depends to a large extent on the *cost model*; that is, how the cost incurred by an individual agent traversing an edge is affected by the congestion (i.e., the number of agents) on that edge. We consider three natural cost models that correspond to different congestion effects.

The first is the *independent cost model*, where the cost incurred by an agent that traverses an edge equals precisely the edge weight, independent of the number of agents traversing it. This model captures externality-free settings, such as wide roads that are not affected by congestion.

The other two cost models consider congestion effects, which can be either negative or positive. The *routing cost model* captures settings exhibited by *negative congestion effects*, where an individual's cost increases as the number of agents using its resource increases. This situation is typical, for example, in routing applications. The *shared cost model* captures settings exhibited by *positive congestion effects*, where an individual's cost decreases as the number of agents using its resource increases. This situation is typical, for example, in settings where resources are associated with some maintenance cost that should be covered by its users. The exact formulation of these two models is presented in the next section, where the formal model is established.

1.1. The Model

Consider a graph $G = (V(G), E(G))$ that may be directed or undirected, whose topology is fully known. Let s and t be two designated vertices in $V(G)$ and suppose that every edge in $E(G)$ appears on some simple (s,t)-path in G. Unless stated otherwise, all graphs considered in this paper are assumed to satisfy this property. We shall use the convention that $n = |V(G)|$ and $m = |E(G)|$.

Each edge $e \in E(G)$ is associated with some non-negative *weight* $w(e)$; these edge weights are *unknown* a priori. Information on the edge weights is gathered by means of *agents* that explore the graph. Each agent i traverses some (not necessarily simple) (s,t)-path P_i in G and reports the *cost* $c(P_i)$ of P_i. The exact definition of this cost will soon be clarified, but first, let us introduce the following notation: Define $\alpha(e,i)$ to be the number of appearances of edge e in P_i (recall that e may appear multiple times in P_i as P_i is not necessarily a simple path) and fix $\alpha(e) = \sum_i \alpha(e,i)$.

We now turn to define the cost $c(P_i)$ of path P_i. We consider three different *cost models*:

Independent costs. The cost of P_i under the independent cost model is defined simply as the *length* of P_i with respect to the edge weights, that is, $c(P_i) = \sum_{e \in E(G)} \alpha(e,i) \cdot w(e)$.

Routing costs. This cost model captures settings exhibited by *negative congestion effects*, where agents suffer from using congested resources. In particular, in this model, the cost incurred by agent i for using edge e increases linearly with $\alpha(e)$. Formally, under the routing cost model, the cost of P_i is defined as $c(P_i) = \sum_{e \in E(G)} \alpha(e,i) \cdot \alpha(e) \cdot w(e)$.

Shared costs. This cost model captures settings exhibited by *positive congestion effects*, where agents benefit from having other agents sharing their resources. In particular, in this model, the weight of an edge is assumed to be shared equally by all agents using this edge. Formally, the cost of P_i is defined as $c(P_i) = \sum_{e \in E(G), \alpha(e) > 0} \frac{\alpha(e,i)}{\alpha(e)} \cdot w(e)$.

Depending on the cost model, the agents can provide crucial information on the edge weights by exploring various (s,t)-paths in G. This can be employed to implement:

(a) a *weight discovery protocol* whose goal is to identify the weights of all edges in $E(G)$; and

(b) a *shortest path discovery protocol* whose goal is to identify some shortest (s,t)-path in G.

The discovery protocols organize their agents in *rounds*. This has two implications. First, the paths traversed by the agents of round $t+1$ are decided only after the agents of round t have made their reports. In particular, this means that the protocol admits a certain level of adaptivity. Second, the variables $\alpha(e,i)$ and $\alpha(e)$ are determined for each round independently of other rounds. In other words, agents belonging to disjoint rounds do not "interfere" with each other. This does

		Independent	Shared/Routing
Undir.	**Weight**	≤ 1 round, m agents [5] $\geq m$ agents [1]	≤ 1 round, m agents [5] $\geq m$ agents [1]
	SP	≤ 1 round, m agents [5] $\geq m-n+1$ agents [7]	≤ 1 round, m agents [5] $\geq m-n+1$ agents [7]
DAG	**Weight**	Impossible [4]	$\leq m$ rounds, $m+n-2$ agents (1-2 per round) [2] ≤ 2 rounds (in-degree ≥ 2) [3] $\geq m$ agents [1]
	SP	≤ 1 round, $m-n+2$ agents [9] $\geq m-n+1$ agents [6]	$\leq m-n+2$ rounds, 1 agent per round [9] ≤ 2 rounds (in-degree ≥ 2) [3] ≥ 2 rounds (existential) [8] $\geq m-n+1$ agents [6]

Table 1 Summary of our results. The columns correspond to the different cost models, and the rows correspond to either undirected or directed acyclic graphs (DAG). The rows are further divided into weight discovery (Weight) or shortest path discovery (SP) protocols. The cells specify the bounds on the number of agents and rounds, expressed in terms of the number of edges (m) and the number of vertices (n); the relevant theorem number appears in brackets.

not affect the independent cost model; however, it may be crucial for discovery protocols operating under the shared and routing cost models.

The cost of a discovery protocol is typically measured by the number of rounds and the number of agents used within each round. Therefore, a natural objective of a discovery protocol is to minimize these two values.

1.2. Our Results

Our results are summarized in Table 1. The first set of results concerns undirected graphs. We show that a single round is always sufficient to discover all edge weights regardless of the cost model. As for the number of agents, we establish an upper bound of m (Theorem 5), which is easily shown to be tight (Theorem 1). We also show that the bound of m agents can be achieved by traversing paths that resemble simple paths very closely; in particular, the only deviation of an agent from a simple path is that it may need to traverse a single edge on the simple path back and forth once beyond the first traverse. For the shortest path discovery problem, we establish a slightly weaker lower bound of $m-n+1$ on the number of agents, again, regardless of the cost model (Theorem 7).

We proceed with the results for directed acyclic graphs (DAGs), beginning with two simple observations: (1) Under the independent cost model, only weights of direct (s,t)-edges can be determined (Theorem 4). (2) Unlike undirected graphs, a single round may not be sufficient to determine a shortest path under the shared and routing cost models (Theorem 8). As we explain later on, under the shared and routing cost models, weight discovery in DAGs is only possible if there is no pair of edges that appear together in every path (recall that under the independent cost model, weight discovery in DAGs is impossible regardless); assuming that a given DAG does not admit such a pair of edges, we establish the existence of a weight discovery protocol that operates in m rounds with $m+n-2$ agents (Theorem 2). We also show that under the shared and routing cost models, if all internal vertices of a DAG have in-degree at least 2, then weight discovery is possible in 2 rounds, although this requires (possibly exponentially) many agents (Theorem 3). This bound is tight as demonstrated by the example given in Theorem 8. Shortest path discovery protocols in DAGs can be implemented in at most $m-n+2$ rounds with at most that many agents under all cost models (Theorem 9). On the negative side, such discovery protocols require at least $m-n+1$ agents (Theorem 6); this also holds under all cost models.

Note that with the exception of the lower bound stated in Theorem 8, all our lower bounds hold for all, or almost all, graph instances. Our techniques combine tools from linear algebra with results in graph theory, including (s,t)-numbering of biconnected graphs and facts about the flow space of directed graphs, together with some simple probabilistic arguments.

1.3. Related Work

The problems considered here are similar to those of hidden graph reconstruction, where the goal is to reconstruct a hidden graph using small number of queries. In particular, the basic problem is as follows: given a hidden weighted or unweighted graph from a prescribed family of graphs, the objective is to identify the graph (and the weights of its edges, in the weighted case), by asking a small number of queries. A typical query in the basic model is to check whether or not a given set of vertices contains at least one edge of the graph. An additive query returns the sum of weights of all the graph's edges in this set. The algorithms considered can be adaptive or non-adaptive. Similar questions have been considered in the literature on Group Testing (see Ainger 1998, Du and Hwang 1993), but the variants dealing with graphs are more recent and arise in the study of questions in computational biology. Here the vertices correspond to molecules, the edges to reactions between pairs of them, and the queries correspond to experiments of putting a set of molecules together in a test tube and determining whether a reaction occurs (or how many reactions occur, in the additive model). See Grebinski and Kucherov (2000), Alon et al. (2002), Alon and Asodi (2005), Bouvel et al. (2005), Angluin and Chen (2006), Choi and Kim (2008), Bshouty and Mazzawi (2009) and the references therein for the known results on these questions.

The problems addressed in the present paper are related to these results, and especially to the ones dealing with the additive model in the weighted case. The main differences are that each query here corresponds to the set of edges of an (s,t)-path, and not to the collection of all edges in a given set of vertices, and that the underlying graph here is known and only the weights (or the edges and total weight of a shortest (s,t)-path) have to be determined. The techniques in most of the papers dealing with hidden graphs rely mainly on probabilistic ideas, and do not share much with the tools applied here, besides the obvious connection to linear algebra.

Problems in this spirit have been also studied in the networking literature, with the goal of discovering the underlying network topology using limited and local information. One of the approaches for topology discovery is called network tomography, where topology inference is carried out by end-to-end packet probing measurements (such as delays). The goal is to design algorithms in which the number of measurements required is small. See Castro et al. (2004), Duffield et al. (2000), Beerliova et al. (2006) and references therein for more details on network tomography and topology discovery in particular. Apart from the different types of queries and the different flavor of the analysis, the focus in this literature is on learning the topology of the graph, whereas our goal in the current paper is to learn the edge weights (or a shortest path with respect to those).

The cost models used in this paper are all special cases of the congestion game model (Rosenthal 1973, Monderer and Shapley 1996), where the cost of each resource is a function of its congestion; i.e., the number of agents using that resource. In particular, let w denote an edge's weight, x denote an edge's congestion, and $f(x)$ denote the resource's cost function, then the independent cost model corresponds to a constant cost of $f(x) = w$, the routing cost model corresponds to a linear cost of $f(x) = xw$, and the shared cost corresponds to the function of $f(x) = w/x$. Congestion games were extensively studied from a game theoretic perspective, considering both negative congestion effects, as in routing (Roughgarden and Tardos 2002), and positive congestion effects, as in fair cost-sharing (Anshelevich et al. 2008). However, our model is different in two fundamental ways. First, unlike a congestion game, where the resources' costs are typically known, here, the challenge is to discover these costs. Second, the agents in our case are assumed to be obedient.

Finally, our work is also related to the work by Papadimitriou and Yannakakis (1991), which deals with a setting where the graph is dynamically specified. Due to the dynamic revelation, the shortest-path cannot always be discovered, and the goal is to find strategies that optimize the worst-case ratio of the distance covered to the length of the shortest path.

1.4. Paper structure

In Section 2 we establish an algebraic view of our setting, along with some additional notations. The algebraic view and notations will be used in our analysis throughout the paper. Section 3 studies weight discovery protocols, and shortest-path discovery protocols are studied in Section 4.

2. Preliminaries

Consider some graph G that may be directed or undirected. The vertex set and edge set of G are denoted $V(G)$ and $E(G)$, respectively. We use the standard notions of (simple) *path* and *cycle*. In general, we treat a path P as a multiset of edges and denote the set of vertices along P by $V(P)$. Given two vertices $u, v \in V(G)$, a (u, v)*-path* is a path that leads from u to v. If P is a (u_1, u_2)-path and P' is a (u_2, u_3)-path, then $P \circ P'$ refers to the (u_1, u_3)-path formed by the *concatenation* of P and P'.

Unless stated otherwise, the graph G is assumed to admit some designated vertices s and t so that every edge in $E(G)$ appears on some simple (s, t)-path in G. The parameters m and n are used to denote $|E(G)|$ and $|V(G)|$, respectively.

Discovery protocols — an algebraic view.

A single round of a discovery protocol is depicted by a real matrix M with k rows and m columns. (Recall that $m = |E(G)|$; the parameter k is chosen by the protocol.) Each row $1 \leq i \leq k$ of M corresponds to some (not necessarily simple) (s, t)-path P_i in G. Assuming that edge e_j appears $\alpha(j, i)$ times in P_i, and fixing $\alpha(j) = \sum_i \alpha(j, i)$, we have

$$M_{i,j} = \begin{cases} \alpha(j,i), & \text{under the independent cost model;} \\ \frac{\alpha(j,i)}{\alpha(j)}, & \text{under the shared cost model;} \\ \alpha(j,i) \cdot \alpha(j), & \text{under the routing cost model.} \end{cases}$$

A matrix M that can be constructed in that manner is called *attainable*.

In each round t, the discovery protocol chooses some attainable matrix M^t; in return, it obtains the vector $M^t \vec{w}$, where $\vec{w} \in \mathbb{R}^m_{\geq 0}$ is the vector of unknown edge weights. This new information can be used to design the attainable matrices $M^{t'}$ of rounds $t' > t$. The rows of the attainable matrix M^t are referred to as the *queries* of round t; the vector $M^t \vec{w}$ is referred to as the vector of *answers* of round t. Given some collection $\mathcal{M}$ of attainable matrices, we say that the vector $\vec{v} \in \mathbb{R}^m$ is *spanned* by $\mathcal{M}$ if it is spanned by the union of the rows of matrices in $\mathcal{M}$.

Given some edge subset $F \subseteq E(G)$, we denote the characteristic vector of F by $\chi(F)$, i.e., $\chi(F)_j = 1$ if $e_j \in F$; and $\chi(F)_j = 0$ otherwise. This definition is extended to multisets in the natural way. If F is a singleton, namely, $F = \{e\}$, then we may slightly abuse the notation and write $\chi(e)$ instead of $\chi(\{e\})$. The *path space* of G is the vector space spanned by the characteristic vectors of the paths of G. The *(s, t)-path space*, *simple* (s, t)*-path space*, and *cycle space* of G are defined in the same manner.

In light of this definitions, the goal of a weight discovery protocol is to span the vectors $\chi(e)$ for all edges $e \in E(G)$; the goal of a shortest path discovery protocol is to identify some (s, t)-path P such that $\chi(P) \cdot \vec{w} \leq \chi(P') \cdot \vec{w}$ for every (s, t)-path P' in G.

3. Weight Discovery

In this section we study the weight discovery problem in both undirected graphs and DAGs. As a first (warm up) step, we observe that regardless of the cost model, any weight discovery protocol requires at least m agents.

THEOREM 1. *Let G be an arbitrary undirected graph or DAG. Any weight discovery protocol for G requires at least m agents, regardless of the cost model.*

Proof. Since the protocol should span the m independent vectors $\chi(e)$, $e \in E(G)$, it follows that the total number of queries (i.e., the total number of rows in the attainable matrices used by the protocol) must be at least m. □

3.1. Upper Bound: General DAGs

Consider some DAG G with a single source s and a single sink t.[1] We say that G is *edge distinguishable* if for every two edges $e, e' \in E(G)$, there exists some (s,t)-path P in G such that exactly one of the edges e and e' appears in P. Note that if there exist two edges $e, e' \in E(G)$ such that for every (s,t)-path P in G, either $\{e, e'\} \subseteq P$ or $\{e, e'\} \cap P = \emptyset$, then every vector $\vec{v} \in \mathbb{R}^m$ that can be spanned by the queries must satisfy $\vec{v}(e) = \vec{v}(e')$, which, in particular, means that the characteristic vectors of e and e' cannot be spanned. Therefore, $w(e)$ and $w(e')$ cannot be discovered and G is not suitable for a weight discovery protocol. Our goal in this section is to establish the following theorem.

THEOREM 2. *Every edge distinguishable DAG admits a weight discovery protocol that operates in m rounds with $m+n-2$ agents (in total) under the shared and routing cost models.*

The edge distinguishability assumption can be lifted if one is interested in a shortest path discovery protocol, rather than a weight discovery protocol; from a technical perspective, implementing the former turns out to be a much easier task (see Theorem 9).

Outline. Theorem 2 is established in four stages:

(i) We introduce the notion of an *edge distinguishable* path and show that every path in G is edge distinguishable.

(ii) We argue that if P is an edge distinguishable path, then there must exist an edge $e \in P$ whose weight $w(e)$ can be determined.

(iii) We observe that the path P' obtained from P by contracting the edge e is edge distinguishable in the corresponding directed graph G' (which is not necessarily acyclic anymore). Then, we can continue by induction and discover the weights of all edges in P.

(iv) The edge weight exploration process derived from steps (ii) and (iii) may require "too many" rounds and agents. However, we identify a basis for the row spaces of all the attainable matrices involved in the process and show that this basis can be spanned in m rounds with $m+n-2$ agents.

Bypassing edge distinguishable paths. Fix an arbitrary (not necessarily acyclic) directed graph G with two designated vertices $s, t \in V(G)$. We start with the following observation.

OBSERVATION 1. Consider two simple (s,t)-paths P, P' in G. Then the vectors $\chi(P \cap P')$, $\chi(P - P')$, and $\chi(P' - P)$ can be spanned in 2 rounds with 1 agent in the first round traversing P and 2 agents in the second round, one traversing P and the other traversing P'.

Consider some simple (s,t)-path P in G. A simple (u,v)-path Q in G is referred to as a (u,v)-*bypass* of P if (1) $V(Q) \cap V(P) = \{u, v\}$; (2) u precedes v along P and (3) $E(Q) \cap E(P) = \emptyset$. The bypass Q induces a partition of P into three disjoint paths: the (possibly empty) (s,u)-subpath $P_{s,u}$, the (u,v)-subpath $P_{u,v}$, and the (possibly empty) (v,t)-subpath $P_{v,t}$. The simple (s,t)-path $P_{s,u} \circ Q \circ P_{v,t}$ is referred to as a *detour* of P. We say that a vector in $\mathbb{R}^m$ can be *P-spanned* if it can be spanned by agents traversing the path P and its detours. The following observation is a direct consequence of Observation 1.

OBSERVATION 2. The vectors $\chi(Q)$, $\chi(P_{u,v})$, and $\chi(P_{s,u} \cup P_{v,t})$ can be P-spanned in 2 rounds with 3 agents (in total).

The simple path P is said to be *edge distinguishable* if for every two edges $e, e' \in P$, $e \neq e'$, there exists some (u,v)-bypass Q of P such that exactly one of the edges e and e' appears in $P_{u,v}$; in that case we say that the bypass Q *distinguishes* between e and e' in P. The following lemma can now be established.

LEMMA 1. *Consider some directed graph G with two designated vertices $s, t \in V(G)$ and let P be an edge distinguishable (s,t)-path in G. Then there exists some edge $e \in P$ such that $\chi(e)$ can be P-spanned.*

Proof. Let $P = (v_0, \ldots, v_k)$, where $s = v_0$ and $t = v_k$. To avoid cumbersome notation, we may identify the vertex v_i with the integer i when the reference to the actual vertex is clear from the context. The assertion holds trivially if P consists of a single edge, so in what follows we assume that $|P| = k \geq 2$. Since P is edge distinguishable, there must exist some bypass Q that distinguishes between the first and last edges in P.

Let (i^*, j^*), $0 \leq i^* < j^* \leq k$, be the pair that minimizes $j^* - i^*$ among all pairs (i, j), $0 \leq i < j \leq k$, that satisfy:

- $\chi(P_{i,j})$ can be P-spanned;
- P admits an (i', j)-bypass for some $0 \leq i' \leq i$; and
- P admits an (i, j')-bypass for some $j \leq j' \leq k$.

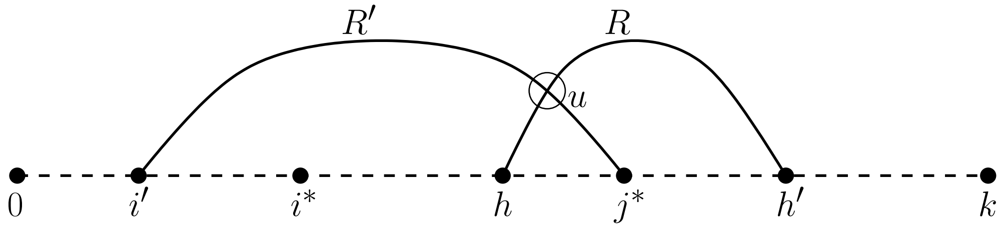

Figure 1 The path P (dashed line) and the bypasses R and R' (solid curves).

This is well defined as Observation 2 guarantees that the start vertex and end vertex of the aforementioned bypass Q are candidates for i^* and j^*, respectively.

We argue that $j^* - i^* = 1$. The assertion follows as this implies that (i^*, j^*) is an edge whose characteristic vector can be P-spanned. Assume towards deriving a contradiction that $j^* - i^* > 1$ and consider the (disjoint) edges $e = (i^*, i^* + 1)$ and $e' = (j^* - 1, j^*)$. Since P is edge distinguishable, and in particular, e and e' can be distinguished, there must exist some bypass R of P such that exactly one of its endpoints is in the interval $[i^* + 1, j^* - 1]$ — denote this endpoint by h. Assume without loss of generality that the other endpoint of R is $j^* \leq h' \leq k$. Recall that by the definition of the pair (i^*, j^*), there exists some (i', j^*)-bypass R' of P such that $0 \leq i' \leq i^*$. Refer to Figure 1 for illustration.

We consider two possible cases. Assume first that $R \cap R' \neq \emptyset$ and let u be a vertex which is internal to both R and R'. (Such a vertex u must exist if R and R' share an edge.) Consider the path R'' that starts at h, follows R until u, and then follows R' until j^*. By definition, R'' serves as an (h, j^*)-bypass of P. Observation 2 guarantees that $\chi(P_{h,j^*})$ can be P-spanned (due to the bypass R''). But this contradicts the choice of the pair (i^*, j^*): the pair (h, j^*) should have been chosen due to the bypasses R' and R.

So, assume that $R \cap R' = \emptyset$. By applying Observation 1 to the detours $P_{0,i'} \circ R' \circ P_{j^*,k}$ and $P_{0,h} \circ R \circ P_{h',k}$ of P, we conclude that the characteristic vector of $P_{0,i'} \cup P_{h',k}$, which is the intersection of the two detours, can be P-spanned. This implies that $\chi(P_{i',h'}) = \chi(P) - \chi(P_{0,i'} \cup P_{h',k})$ can also be P-spanned. Since Observation 2 guarantees that $\chi(P_{h,h'})$ can be P-spanned (due to the bypass R) and that $\chi(P_{i',j^*})$ can be P-spanned (due to the bypass R'), it follows that $\chi(P_{h,j^*}) = \chi(P_{h,h'}) + \chi(P_{i',j^*}) - \chi(i', h')$ can also be P-spanned. But this contradicts the choice of the pair (i^*, j^*): the pair (h, j^*) should have been chosen due to the bypasses R' and R. Therefore, $j^* - i^* = 1$ and the assertion holds. □

Next, we show that Lemma 1 can be employed to P-span the characteristic vectors of all edges in the edge distinguishable path P.

Lemma 2. *Consider some directed graph G with two designated vertices $s, t \in V(G)$ and let P be an edge distinguishable (s, t)-path in G. Then $\chi(e)$ can be P-spanned for all edges $e \in P$.*

Proof. The assertion holds trivially if P consists of a single edge, so assume that $|P| \geq 2$. Lemma 1 guarantees that $\chi(e)$ can be P-spanned for some edge $e = (u, v) \in P$, hence we may assume that we already know its weight $w(e)$. Let G' be the directed graph obtained from G by contracting the vertices u and v into a single vertex z and let $P' = P - \{e\}$ be the path in G' obtained from P by this contraction. Since P is an edge distinguishable path in G, it follows that P' is an edge distinguishable path in G'. The assertion is established by showing that queries corresponding to the path P' and its detours under G' can be simulated by queries corresponding to the path P and its detours under G, thus we can apply Lemma 1 to G' and continue by induction[2] on $|P|$.

Since $P' = P - \{e\}$ and since we already know $w(e)$, it follows that we can translate the answer of a query corresponding to P under G to the answer of a query corresponding to P' under G'. Consider some detour Q' of P' in G' and let R be the corresponding bypass. If the edges of Q' form a path in G, then a query corresponding to Q' under G' is also a query under G. So assume that the edges of Q' do not form a path in G, which means that Q' must go through the vertex z. Since the contracted edge e belongs to the path P, the bypass R of P' in G' (a subpath of Q') is also a bypass of P in G, thus the detour Q of P in G that corresponds to R satisfies $Q = Q' \cup \{e\}$. As before, since $w(e)$ is already known, we can translate the answer of a query corresponding to Q under G to the answer of a query corresponding to Q' under G'. The assertion follows. □

Now, let G be a DAG with a single source s and a single sink t and suppose that G is edge distinguishable. We argue that the fact that G is a DAG implies that every simple (s, t)-path in G is edge distinguishable. Indeed, if $e, e' \in P$, $e \neq e'$, then a bypass for P that avoids e can be obtained from an (s, t)-path Q such that $e' \in Q$, $e \notin Q$. As every edge of G appears on some simple (s, t)-path in G, Lemma 2 guarantees that the vectors $\chi(e)$ can be spanned for all $e \in E(G)$, that is, we can implement a weight discovery protocol. However, a weight discovery protocol implemented by following the process implicitly described in the proofs of Lemmas 1 and 2 requires too many rounds and agents.

Extending a basis for the cycle space. To tackle this obstacle, we consider the directed graph G' obtained from G by identifying the vertices s and t into a new vertex z. Since s and t are the unique source and sink of the DAG G, it follows that G' is strongly connected and that every cycle in G' includes the vertex z. The following lemma is a special case of Theorem 14.2.1 in Godsil and Royle (2001).[3]

Lemma 3. *The dimension of the cycle space of a strongly connected digraph H is $|E(H)| - |V(H)| + 1$.*

By applying Lemma 3 to G' (that admits $n - 1$ vertices and m edges), we obtain the following corollary.

COROLLARY 1. *The dimension of the (s,t)-path space in G is $m-n+2$. A basis B for this space can be queried in $m-n+2$ rounds, each with a single agent.*

A careful examination of the proofs of Lemmas 1 and 2 leads to the conclusion that they essentially rely on successive applications of the building block established in Observation 1. This building block consists of two rounds, the first with a single agent and the second with two agents. The basis B promised by Corollary 1 spans the queries corresponding to the single agent rounds. This basis B can be extended to a basis for the whole edge space of G by appending to it $m-(m-n+2)=n-2$ additional vectors, and as we have just seen, these additional vectors can be chosen such that each one of them corresponds to a query involving two agents. In total, a basis for the edge space of G is spanned in $m-n+2$ one-agent rounds plus $n-2$ two-agent rounds, which sums up to m rounds with $m+n-2$ agents. Theorem 2 follows.

3.2. Upper Bound: DAGs with In-Degrees ≥ 2

It turns out that weight discovery protocols can be much more efficient in terms of the number of rounds if every internal vertex in the DAG has in-degree at least 2. (The same result holds if every internal vertex has out-degree at least 2 as we can always apply our protocol to the DAG obtained from reversing the orientation of all edges.) Note that in the scope of this section, we do not attempt to bound the number of agents employed by our discovery protocol.

THEOREM 3. *Every DAG for which the in-degree of every internal vertex is at least* 2 *admits a weight discovery protocol that operates in two rounds under the shared and routing cost models.*

Let $\mathcal{P}$ be the set of all (s,t)-paths of G. In each of the two rounds, for every path $P\in\mathcal{P}$, the protocol chooses a random number of agents from a sufficiently large set of numbers and sends them along P. We argue that with probability at least $1/2$, this protocol determines the weights of all edges. Since this probabilistic argument holds for all DAGs promised by Theorem 3, it follows that the edge weights of each such DAG can be discovered (deterministically) in 2 rounds, thus establishing the theorem.

Fix a topological order on the vertices of G, and let v be the first vertex (succeeding s) in the topological order. We begin with the following lemma.

LEMMA 4. *The weights of all the edges from s to v can be determined by the protocol.*

Proof. Since the in-degree of every vertex is at least 2, there are at least two edges from s to v. Let e_1 and e_2 be two (s,v) edges. We show that the weights of e_1 and e_2 can be determined. The same analysis can be applied to any other pair of edges from s to v to establish the assertion of the lemma. Throughout this proof we denote $w_i=w(e_i)$ for simplicity.

Let $\mathcal{P}_1 = \{P : e_1 \in P\}$ and $\mathcal{P}_2 = \{P : e_2 \in P\}$. For every $P \in \mathcal{P}$, let x_P^1 denote the number of agents that traverse the path P in round 1 (which is chosen randomly by the protocol). Also, let $x_1^1 = \sum_{P\in\mathcal{P}_1} x_P^1$ and $x_2^1 = \sum_{P\in\mathcal{P}_2} x_P^1$; i.e., x_1^1 (respectively, x_2^1) is the number of agents traversing paths in $\mathcal{P}_1$ (resp., $\mathcal{P}_2$) in the first round. Since e_1 and e_2 both go from s to v, the sets $\mathcal{P}_1$ and $\mathcal{P}_2$ are disjoint, consequently x_1^1 and x_2^1 are independent, as the sums of independent variables.

Let M^1 and M^2 be the matrices constructed in the first and second rounds of the protocol under the routing cost model. Consider some (v,t)-path $P_{v,t}$, and let $P = e_1 \circ P_{v,t}$ and $P' = e_2 \circ P_{v,t}$. Let M_P^1 (resp., $M_{P'}^1$) denote a row in M^1 corresponding to an agent traversing the path P (resp., P'). The value $(M_P^1 - M_{P'}^1)\vec{w}$ can be computed as the difference of two answers of the protocol; denote it by b^1. Since P and P' share the same suffix, it holds that $(M_P^1 - M_{P'}^1)\vec{w} = x_1^1 w_1 - x_2^1 w_2$. We obtain the equation $x_1^1 w_1 - x_2^1 w_2 = b^1$, where x_1^1, x_2^1, b^1 are known and x_1^1 and x_2^1 are independent.

Similarly, let x_1^2 and x_2^2 denote the number of agents traversing paths in $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, in the second round. Applying the same analysis to the second round gives us the equation $x_1^2 w_1 - x_2^2 w_2 = b^2$ (with b^2 defined analogously to b^1), where x_1^2 and x_2^2 are independent. We obtain a system of two linear equations with two unknowns, w_1 and w_2. It is easy to verify that for a sufficiently large range of possible values for the number of agents traversing each path, the obtained system of equations has a unique solution with probability at least, say $1 - 1/2m$. Consequently the weights of e_1 and e_2 are determined with high probability. Under the shared cost model, the same analysis can be applied by replacing the coefficients x_i^j by $(x_i^j)^{-1}$ (for $i, j \in \{1, 2\}$). □

The last lemma establishes that the weights of all edges from s to v can be determined by the protocol. Let G' be the graph obtained from G by contracting the vertices s and v into a single vertex s'. Since only (s,v) edges are contracted, the obtained graph G' is also a DAG (with source s' and sink t). In addition, it preserves the in-degree ≥ 2 property. Given that we have already discovered the weight of the (s,v)-edges in G, queries under G' can be simulated by queries under G as the answer of a query corresponding to an (s,t)-path in G can be translated to the answer of a query corresponding to an (s',t)-path in G'. Therefore, we can apply Lemma 4 to G' and proceed by induction on the topological order to determine the weights of all edges in $E(G)$. Theorem 3 follows since the total probability of failure is bounded by $\frac{1}{2m} m \leq 1/2$.

Remark: The bound of 2 rounds is tight even for the shortest path discovery problem as demonstrated by Theorem 8.

3.3. Impossibility: DAGs under independent costs

The next theorem establishes the infeasibility of weight discovery protocols in DAGs under the independent cost model.

Theorem 4. *Let G be an arbitrary DAG with at least 3 vertices. There does not exist any weight discovery protocol for G under the independent cost model.*

Proof. Consider some internal vertex v in the DAG. In any vector that can be spanned by paths, the sum of coordinates corresponding to edges entering v equals the sum of coordinates corresponding to edges exiting v. Therefore, the characteristic vector of an edge incident on v can never be spanned. □

3.4. Upper Bound: Undirected Graphs

This section is dedicated to establishing the existence of a weight discovery protocol for undirected graphs that operates in a single round with m agents.

Theorem 5. *Every undirected graph admits a weight discovery protocol that operates in a single round with at most m agents under all cost models.*

Proof. For every edge $e_i \in E(G)$, let P_i be some (s,t)-path that traverses the edge e_i a "large" number of times x_i (by traversing e back and forth as needed), while traversing every other edge e_j at most once. The exact value of x_i shall be determined soon. The protocol operates in a single round — for every edge e_i, it sends a single agent along the path P_i. We denote by c_i the total number of appearances of the edge e_i in all paths P_j for $j \neq i$. Since for every $i \neq j$ the edge e_i appears in the path P_j at most once, c_i cannot exceed $m-1$.

The matrix M constructed by the protocol has m rows, each corresponding to one of the m agents (or paths). We next specify the values of the matrix entries under the different cost models. Under the independent cost model, for every column i we have $M_{i,i} = x_i$, and for every $j \neq i$, $M_{j,i} = 1$ if e_i appears in P_j and $M_{j,i} = 0$ otherwise. Since for every column i there are exactly c_i non-zero entries off the diagonal, it follows that $\sum_{j \neq i} M_{j,i} = c_i$. Consequently, by choosing $x_i > m-1 \geq c_i$ for every i, the matrix M is (strictly) diagonally dominant and therefore non-singular (see, e.g., Horn and Johnson 1985). In particular, it spans the vectors $\chi(e_i)$ for all edges $e_i \in E(G)$.

A similar analysis shows that in the shared and routing cost models the obtained matrix is also diagonally dominant if $x_i > m-1 \geq c_i$ for every i. Specifically, for every column i, under the shared cost model, we have $M_{i,i} = x_i/(x_i + c_i)$ and $M_{j,i} \in \{1/(x_i + c_i), 0\}$ for $j \neq i$, whereas under the routing cost model, we have $M_{i,i} = x_i(x_i + c_i)$ and $M_{j,i} \in \{x_i + c_i, 0\}$ for $j \neq i$. The assertion holds since in both cases, column i contains exactly c_i non-zero off-diagonal entries. □

The above protocol discovers the weights of all edges and is tight in terms of the number of agents (and obviously in the number of rounds). One may wonder, however, whether it is possible to improve the number of edge traverses in the protocol. We answer this question in the affirmative. Specifically, we argue that if the protocol chooses x_i randomly and uniformly to be either 1 or 3 for every i, then with positive probability, the weights of all edges can be determined. Since this probabilistic argument holds for all undirected graphs, it follows that the edge weights of each undirected graph can be discovered (deterministically) in a single round with agents that traverse "almost" simple paths.

Let M be a random matrix that is constructed in the independent cost model by choosing x_i randomly and uniformly to be either 1 or 3. The determinant of M is a multilinear polynomial of $x_1, \ldots, x_m$. It is easy to verify that a multilinear polynomial of m variables that is not identically zero, whose variables are chosen randomly and uniformly from a set of cardinality 2, is non-zero with positive probability (see, e.g., Alon 1999, Lemma 2.1). It follows that with positive probability, M is non-singular, and thus it spans the characteristic vectors of all edges.

The same analysis can be applied to the shared and routing cost models by defining $w'_i = w_i/(x_i + c_i)$ and $w'_i = w_i(x_i + c_i)$ (where $w_i = w(e_i)$), respectively, and observing that in both cases multiplying the constructed matrix by $\vec{w}$ is equivalent to multiplying the matrix M (constructed in the independent cost model) by $\vec{w}'$.

4. Shortest Path Discovery

In this section we study the shortest path discovery problem, starting with a lower bound for DAGs.

4.1. Lower Bound: DAGs

THEOREM 6. *Let G be an arbitrary DAG. Any shortest path discovery protocol for G requires at least $m - n + 1$ agents, regardless of the cost model.*

Proof. Suppose by way of contradiction that some shortest path P has been discovered with fewer than $m - n + 1$ agents, and let M be the matrix representing the queries of the discovery protocol. Note that M is not necessarily an attainable matrix, rather it may be a row concatenation of several attainable matrices.

The heart of the proof relies on the following adversary argument. An adversary keeps answering the queries as if all (s,t)-paths of G have exactly the same length. To see that this is possible, fix some topological order on the vertices of G and associate every edge e with a weight p_e that equals the difference between the ranks of its incident vertices in the topological order. This way, p_e is strictly positive for every e and the length of every (s,t)-path is $n - 1$. We shall next show that the information that was gathered by the discovery protocol is not sufficient to determine a shortest path. Specifically, we will show that one cannot certify that $(\chi(P) - \chi(P'))\vec{w} \leq 0$ for every (s,t)-path P' of G.

Let B be a maximal set of linearly independent characteristic vectors of (s,t)-paths of G, and let $\mathcal{C} = span(\{\chi(P) - \chi(P') : \chi(P') \in B\})$. Since $|B|$ is known to be $m-n+2$ (see Corollary 1) and P is an (s,t)-path of G, the dimension of $\mathcal{C}$ is $m-n+1$. By the contradiction hypothesis, at most $m-n$ agents are employed, thus the dimension of the row space of M is at most $m-n$. It follows that there must exist a vector $\vec{d} \in \mathcal{C}, \vec{d} \neq 0$, such that $M\vec{d} = 0$. Therefore, for every solution $\vec{p} \in \mathbb{R}^m_{>0}$ that is consistent with the agents' responses (recall that according to our model, all edge weights are positive), there exists a sufficiently small $\epsilon > 0$, such that both $\vec{p} + \epsilon\vec{d}$ and $\vec{p} - \epsilon\vec{d}$ are consistent with the agents' responses (we take ϵ to be sufficiently small so that both $\vec{p} - \epsilon\vec{d}$ and $\vec{p} + \epsilon\vec{d}$ stay positive in all coordinates). Since $\vec{d} \in \mathcal{C}, \vec{d} \neq 0$, there must exist an (s,t)-path P' satisfying

$$\vec{d}(\chi(P) - \chi(P')) \neq 0\,. \tag{1}$$

To conclude the proof we show that the sign of $(\chi(P) - \chi(P'))\vec{w}$ cannot be determined.

To that end, note that since all the queries have been answered as if all path lengths are equal, it follows that $\vec{p}(\chi(P) - \chi(P')) = 0$. Combined with Equation 1 we get that

$$\begin{aligned} & (\vec{p} + \epsilon\vec{d})(\chi(P) - \chi(P')) \\ = \; & \vec{p}(\chi(P) - \chi(P')) + \epsilon\vec{d}(\chi(P) - \chi(P')) \\ = \; & +\epsilon\vec{d}(\chi(P) - \chi(P'))\;, \end{aligned}$$

while

$$\begin{aligned} & (\vec{p} - \epsilon\vec{d})(\chi(P) - \chi(P')) \\ = \; & \vec{p}(\chi(P) - \chi(P')) - \epsilon\vec{d}(\chi(P) - \chi(P')) \\ = \; & -\epsilon\vec{d}(\chi(P) - \chi(P'))\;. \end{aligned}$$

But since both $\vec{p} + \epsilon\vec{d}$ and $\vec{p} - \epsilon\vec{d}$ belong to the solution space, it follows that the sign of $(\chi(P) - \chi(P'))\vec{w}$ cannot be determined, contradicting the assumption that P has been discovered as a shortest path. □

4.2. Lower Bound: Undirected Graphs

A close examination of the proof of Theorem 6 reveals that the same analysis can be applied to an undirected graph G if one can show that G can be directed and assigned positive edge weights so that all of the (s,t)-paths of the obtained graph have the same length. This argument is established in the following lemma, using the properties of st-numbering of graphs, introduced in Lempel et al. (1967).

LEMMA 5. *Let G be an undirected graph with two designated vertices $s,t \in V(G)$ so that every edge appears on some simple (s,t)-path. The edges of G can be directed and assigned positive edge weights such that the length of every (s,t)-path of the obtained graph equals $n-1$.*

Proof. Partition the edges of G into maximal biconnected components (blocks) $C_1,\dots,C_k$. Let $T(G)$ be the *block tree* associated with G (that is, associate a vertex in $T(G)$ with every block of G, and connect two vertices of $T(G)$ by an edge if their corresponding blocks share a vertex in G). It is easy to verify that since every edge appears on some (s,t)-path in G, $T(G)$ forms a simple path. Rename the blocks so that $s \in V(C_1)$, $t \in V(C_k)$, and the incident vertices of every edge in $T(G)$ correspond to blocks C_i and C_{i+1} for some $1 \le i \le k-1$. Let $v_{i,i+1} \in V(G)$ denote the unique vertex that is shared by blocks C_i and C_{i+1} for $1 \le i \le k-1$. Let $s_i = v_{i-1,i}$ be the source of block C_i for every $2 \le i \le k$, and $t_i = v_{i,i+1}$ be the target of block C_i for every $1 \le i \le k-1$, so that $t_{i-1} = s_i$ for every $2 \le i \le k-1$. In addition, let $s_1 = s$ and $t_k = t$ be the source of C_1 and the target of C_k, respectively.

Consider the block C_1. Since it is a biconnected component, it admits an s_1t_1-numbering (Lempel et al. 1967), i.e., an assignment of numbers to its vertices so that s_1 has number 1, t_1 has number $|V(C_1)|$, and every other vertex has a neighbor with a smaller number and a neighbor with a larger number. Compute some s_1t_1-numbering of C_1 and let $num(v)$ denote the s_1t_1-numbering of vertex v. Direct every edge in $E(C_1)$ toward the vertex with the larger s_1t_1-numbering. Given an edge $e=(u,v) \in E(C_1)$, assign it weight $w(e) = num(v) - num(u)$. It is easy to verify that with these edge weights the length of every (s_1,t_1)-path in C_1 is exactly $|V(C_1)|-1$.

Now, for every component C_i, $2 \le i \le k$, repeat the aforementioned process by computing an s_it_i-numbering for the vertices in $V(C_i)$ so that $num(s_i) = num(t_{i-1})$ and $num(t_i) = |\bigcup_{j\le i} V(C_j)|$. One can easily verify that the resulting graph is a DAG in which all (s,t)-paths have length n-1. □

Since every path in the graph obtained by the process described in the last lemma is also a path in G, the analysis of the proof of Theorem 6 can be employed to establish the following theorem.

THEOREM 7. *Let G be an arbitrary undirected graph. Any shortest path discovery protocol for G requires at least $m-n+1$ agents, regardless of the cost model.*

Remark: The lower bound obtained in this section is based on algebraic arguments that hold for every matrix that is composed of at most $m-n$ rows. Specifically, it also holds for agents that can traverse any subset of edges, not necessarily ones corresponding to an (s,t)-path.

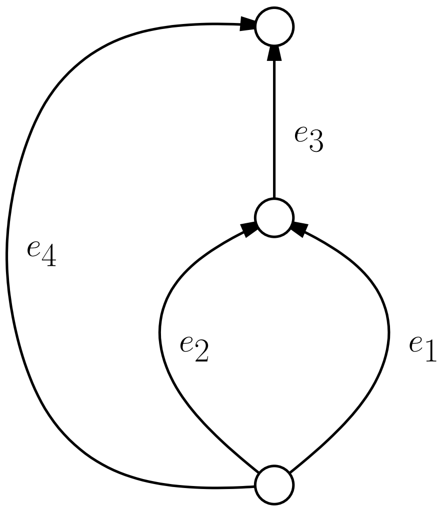

Figure 2 An example of a DAG that does not admit a shortest path discovery protocol operating in a single round under the shared and routing protocols.

4.3. Impossibility: DAGs in a single round

We next show that under the shared and routing cost models, a single round is not always sufficient to determine a shortest path.

THEOREM 8. *There exists a DAG that does not admit any shortest path discovery protocol operating in a single round under the shared or routing cost models.*

Proof. Consider the graph G depicted in Figure 2. G has three (s,t)-paths, namely $P_1 = \{e_1, e_3\}, P_2 = \{e_2, e_3\}$ and $P_3 = \{e_4\}$. We show that under the routing and shared cost models no protocol can discover a shortest path in G with fewer than two rounds.

Consider the routing cost model first. Let x_1, x_2, and x_3 be the number of agents sent by the protocol along the paths P_1, P_2, and P_3, respectively. The answers to the queries corresponding to P_1, P_2, and P_3 are $w_1x_1 + w_3(x_1 + x_2)$, $w_2x_2 + w_3(x_1 + x_2)$, and w_4x_3, respectively. It is trivial to notice that a shortest path cannot be identified if either $x_1 = 0$, $x_2 = 0$, or $x_3 = 0$, so assume hereafter that $x_1, x_2, x_3 > 0$. Clearly, the weight w_4 of the direct edge e_4 can be determined and suppose that $w_4 = 2$. Furthermore, suppose that the answers to the queries corresponding to P_1 and P_2 are consistent with the assignment $w_1 = w_2 = w_3 = 1$ (so that in particular, all three paths have the same length).

Let

$$\vec{u} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \end{pmatrix} \qquad \text{and} \qquad \vec{d} = \begin{pmatrix} -x_2(x_1+x_2) \\ -x_1(x_1+x_2) \\ x_1x_2 \\ 0 \end{pmatrix}.$$

It is easy to verify that the vectors $\vec{d}^+ = \vec{u} + \epsilon\vec{d}$ and $\vec{d}^- = \vec{u} - \epsilon\vec{d}$ are consistent with the answers to all queries and by taking $\epsilon > 0$ to be sufficiently small, we guarantee that both $\vec{d}^+$ and $\vec{d}^-$ are positive in all coordinates, so they are valid solutions for the obtained system of equations.

Now, consider the lengths of the paths P_1, P_2, and P_3 according to the solutions $\vec{d}^+$ and $\vec{d}^-$. According to $\vec{d}^+$, their respective lengths are $2 - \epsilon x_2^2$, $2 - \epsilon x_1^2$, and 2, so either P_1 or P_2 is a shortest-path. In contrast, according to $\vec{d}^-$, their respective lengths are $2 + \epsilon x_2^2$, $2 + \epsilon x_1^2$, and 2, so P_3 is a shortest-path. Since after the first round both $\vec{d}^+$ and $\vec{d}^-$ are valid solutions, a shortest path cannot be discovered.

A similar analysis shows that a shortest path in G cannot be discovered within a single round under the shared cost model either. The main difference here is that we set

$$\vec{d} = \begin{pmatrix} -x_1 \\ -x_2 \\ x_1 + x_2 \\ 0 \end{pmatrix},$$

obtaining a length of $2 \pm \epsilon x_2$ and $2 \pm \epsilon x_1$ for P_1 and P_2, respectively, while the length of P_3 is 2. Again, it is impossible to tell whether or not P_3 is a shortest path.

Remark: In fact, in the last example, as long as $x_1 \neq x_2$, it is impossible to compare any pair of paths within a single round. To see this, let $x_2 = yx_1$ for some $y > 0, y \neq 1$. If $y > 1$, then according to $\vec{d}^+$, P_1 is shorter than P_2, whereas according to $\vec{d}^-$, P_2 is shorter than P_1. If $y < 1$ the opposite holds. In any case, one cannot tell which of P_1 or P_2 is shorter.

4.4. Upper Bound: DAGs

By Lemma 3, the (s,t)-path space of a DAG G has dimension $m - n + 2$, hence a basis for the (s,t)-path space of G can be spanned in $m - n + 2$ rounds, each with a single agent.

Theorem 9. *Every DAG admits a shortest path discovery protocol that operates in $m - n + 2$ rounds with a single agent in each round under the shared and routing cost models; and in* 1 *round with $m - n + 2$ agents under the independent cost model.*

5. Discussion

We introduce a new setting of graph discovery under uncertainty, where the goal is to discover the weights of the network edges, or a shortest path between the designated source and destination nodes. While the work on network topology inference has attracted a lot of interest in multiple disciplines, we consider the complementary problem in which the network topology is given and the goal is to discover the edge weights or the shortest path using end-to-end measurements. We consider three different cost models that correspond to different approaches to congestion effects, and derive bounds on the number of rounds and number of agents that are required by a discovery protocol.

Our study leaves some open questions for future research. First, our results show that while the congestion model does not affect the protocol's complexity in the case of undirected graphs, it has a significant effect if the graph is directed. In particular, congestion effects (either positive or negative) enable the discovery of edge weights in directed graphs, a task that is impossible under the independent cost model. It would be interesting to consider additional natural congestion models and study their effects on the complexity of the discovery protocols. Specifically, the two congestion-aware models considered in this study lead to the same bounds on the protocol's complexity. It is unclear to what extent this result can be extended. Second, our analysis focuses on the case in which a single source and a single destination are designated. The design of discovery protocols for multiple destinations simultaneously seems an interesting extension.

Endnotes

1. A DAG has a single source s and a single sink t if and only if every edge appears on some (s,t)-path.

2. We do not assume that the directed graph G' is acyclic, nor did we assume that the directed graph G is acyclic. All we require for the inductive argument to hold is that the path P' is edge distinguishable.

3. In fact, the setting of Godsil and Royle (2001) is somewhat different from ours. Specifically, they consider an undirected graph and show that the dimension of the flow space associated with any orientation of the graph edges is $m-n+c$, where c is the number of connected components of G. The flow space is spanned by vectors corresponding to the (unoriented) cycles in G, where a flow that does not agree with the orientation of some edge is taken to be negative in the corresponding coordinate. When the orientation of G induces a strongly connected directed graph, the flow space as defined in Godsil and Royle (2001) coincides with the cycle space considered in the current paper.

References

M. Aigner, *Combinatorial Search*, John Wiley and Sons, Chichester, 1988.

N. Alon, *Combinatorial Nullstellensatz*, Combinatorics, Probability and Computing 8, 1999, 7–29.

N. Alon and V. Asodi, Learning a Hidden Subgraph, SIAM J. Discrete Math. 18, 2005, 697–712.

N. Alon, R. Beigel, S. Kasif, S. Rudich and B. Sudakov, Learning a Hidden Matching, Proc. of the 43^{th} IEEE FOCS, IEEE (2002) 197-206. Also: SIAM J. Computing 33, 2004, 487-501.

D. Angluin and J. Chen, Learning a Hidden Hypergraph, Journal of Machine Learning Research 7, 2006, 2215-2236.

E. Anshelevich and A. Dasgupta and J. M. Kleinberg and É. Tardos and T. Wexler and T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. In *SIAM J. Comput.* 38(4):1602–1623, 2008.

Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihalak, L. Shankar Ram. Network discovery and verification. *IEEE Journal on Selected Areas in Communications*, 24(12):2168–2181, 2006.

M. Bouvel, V. Grebinski and G. Kucherov, Combinatorial Search on Graphs Motivated by Bioinformatics Applications: A Brief Survey, In: Kratsch, D. (ed.) WG 2005. LNCS, vol. 3787, pp. 16–27. Springer, Heidelberg, 2005.

N. H. Bshouty and H. Mazzawi, Reconstructing weighted graphs with minimal query complexity, Proc. 20th ALT 2009, 97–109.

R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network Tomography: Recent Developments. In *Stat. Sc.*, 19:499-517, 2004.

S. Choi and J. H. Kim, Optimal Query Complexity Bounds for Finding Graphs. Proc. 40th ACM STOC, 2008, 749–758.

D. Du and F. K. Hwang, *Combinatorial group testing and its application*, Series on applied mathematics, vol. 3. World Science, 1993.

N. Duffield, J. Horowitz, F. Presti, and D. Towsley. Multicast topology inference from end-to-end measurements. In *Advances in Performance Analysis*, 3:207-226, 2000.

Grebinski, V., Kucherov, G.: Optimal Reconstruction of Graphs Under the Additive Model, Algorithmica 28, 104–124, 2000.

C. Godsil and G. Royle, *Algebraic Graph Theory*, volume 207 of Graduate Text in Mathematics, Springer, New York, 2001.

R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

A. Lempel, S. Even, and I. Cederbaum, *An algorithm for planarity testing of graphs*, Theory of Graphs: International Symposium, Rosentiehl, P., ed., New York, 1967, Gordon and Breach, 215–232.

D. Monderer and L.S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

C. H. Papadimitriou and M. Yannakakis. *Shortest paths without a map*, Theoretical Computer Science, 84(1), Pages 127–150, 1991.

R.W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

T. Roughgarden and É. Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.

M. Shih and A. Hero. Unicast inference of network link delay distributions from edge measurements. In Proc. of *IEEE ICASSP*, 6: 3421-3424, 2002.