Economical Graph Discovery*

Noga Alon[†] Yuval Emek[‡] Michal Feldman[§] Moshe Tennenholtz[¶]

Abstract

Consider a weighted n-vertex, m-edge graph G with designated source s and destination t. The topology of G is known, while the edge weights are hidden. Our goal is to *discover* either the edge weights in the graph or a shortest (s,t)-path. This is done by means of *agents* that traverse different (s,t)-paths in multiple *rounds* and report back the total cost they incurred. Various *cost models* are considered, differing from each other in their approach to congestion effects. We seek bounds on the number of rounds and the number of agents required to complete the discovery of the edge weights or a shortest path.

A host 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.

1 Introduction

Suppose you are in an unfamiliar environment and wish to learn how long it takes to travel along different roads; alternatively, you may wish to merely identify a shortest way toward some designated destination. To achieve this goal, you may operate a set of *agents* that traverse specified paths from your origin to the destination and report back the total time duration of their trip. This sort of scenarios typically arise in the context of Internet routing, where autonomous systems or end users are aware of the network topology, but are uncertain about the delays on each link. Learning the link delays may be crucial for routing decisions. This can be achieved by transmitting packets along different paths and observing their trip times. How many rounds of transmissions (as

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

[†]Tel-Aviv University, and Microsoft Israel R&D Center.

[‡]ETH Zurich.

[§]School of Business Administration, The Hebrew University of Jerusalem, and Microsoft Israel R&D Center.

[¶]Microsoft Israel R&D Center, and Technion-Israel Institute of Technology.

a function of the parameters of the problem) are required in order to accomplish this task? How many agents are needed? Our work revolves around this type of questions.

1.1 The Model

Consider a graph $G = (V(G), E(G))$ that may be directed or undirected. 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 (the topology of G is fully known). 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. In fact, we consider three different *cost models*:

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

Things become more complicated when one considers congestion effects that may be either negative or positive as exhibited in the following two models.

Shared costs. In this cost model the weight of an edge is shared by all agents using this edge. Formally, under the shared cost model, the cost of P_i is defined as $c(P_i) = \sum_{e \in E(G)} \frac{\alpha(e,i)}{\alpha(e)} \cdot w(e)$.

Routing costs. In this cost 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 $c(P_i) = \sum_{e \in E(G)} \alpha(e,i) \cdot \alpha(e) \cdot w(e)$.

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

<table>
<tr><th></th><th></th><th>Independent</th><th>Shared/Routing</th></tr>
<tr><td rowspan="2">Undirected</td><td>Weight</td><td>≤ 1 round, m agents
$\geq m$ agents</td><td>≤ 1 round, m agents
$\geq m$ agents</td></tr>
<tr><td>SP</td><td>≤ 1 round, m agents
$\geq m-n+1$ agents</td><td>≤ 1 round, m agents
$\geq m-n+1$ agents</td></tr>
<tr><td rowspan="2">DAG</td><td>Weight</td><td>Infeasible</td><td>$\leq m$ rounds, $m+n-2$ agents (1-2 per round)
≤ 2 rounds (in-degree ≤ 2)
$\geq m$ agents</td></tr>
<tr><td>SP</td><td>≤ 1 round, $m-n+2$ agent

$\geq m-n+1$ agents</td><td>$\leq m-n+2$ rounds, 1 agent per round
≥ 2 rounds (existential)
$\geq m-n+1$ agents</td></tr>
</table>

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 contain the bounds on the number of agents and rounds, expressed in terms of the number of edges (m) and the number of vertices (n).

not affect the independent costs model, however, it may be crucial for discovery protocols operating under the shared and routing costs models. The objective of a discovery protocol would typically be to minimize the number of rounds and the number of agents used within each round.

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 5.1). 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. Our upper bound is accompanied by an almost matching lower bound: no shortest path discovery protocol can operate with less than $m-n+1$ agents, regardless of the number of rounds and the cost model (Corollary 4.3). The strength of this lower bound comes from its universality, i.e., it applies to all undirected graphs.

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 (Proposition 4.6). (2) Unlike undirected graphs, a single round may not be sufficient to determine a shortest path under the shared and routing cost models (Proposition 4.4). Note that 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; 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 (Section 3.1). We also show that under the shared and routing cost models, if all vertices of a DAG have in-degree at least 2, then weight discovery is possible in 2 rounds (Theorem 3.6). This bound is tight as demonstrated by the example given in Proposition 4.4. Shortest path discovery protocols in DAGs can be implemented in at most $m-n+2$ rounds under all cost models. On the negative side, such discovery protocols require at least $m-n+1$ agents (this also holds under all cost models).

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 reconstructing a hidden graph by a small number of queries. Problems of this type have been studied in several papers, motivated by questions in Bioinformatics.

The basic problem in these papers is as follows: given a hidden weighted or unweighted graph H from a prescribed family of graphs, our 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 H. An additive query returns the sum of weights of all edges of H in this set. The algorithms considered can be adaptive or non-adaptive. Similar questions have been considered in the literature on Group Testing (see [1], [10]), 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 [12], [4], [3], [7], [5], [9], [8] 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.

The problems considered here are also similar to those considered in the area of reinforcement learning [20, 14]. Specifically they are related to work in multi-agent reinforcement learning [21, 16], and in particular to work on learning in congestion games [6, 22]. In reinforcement learning agents try to optimize their behavior in an a-priori unknown environment by an explore and exploit approach in which they adapt their behavior based on observed feedback. The learning process is restricted by the agents' observation capabilities. Multi-agent learning has been studied in game theory [11] mainly in the context of repeated games. Our work fits into the perspective of playing a repeated game by a set of cooperative agents, but our set of agents is dynamic, and in fact consists of a single master-agent that decides on the actions to be conducted by a pool of agents in each iteration.

Our model of a network is as discussed in the study of congestion games [19, 17]. The unknown parameters are the resource cost functions as discussed in [6, 22]. Agents' observability (a.k.a. the level of monitoring in game theory) is, however, very restrictive in our setting – we are able to see only the total cost incurred by each agent in each iteration. Dealing efficiently with that restrictive structure is a major contribution of our work. Our emphasis is on cooperative discovery rather than on game-theoretic solution concepts. Our work also applies to both directed and undirected graphs, while the related work on reinforcement learning in congestion games dealt only with directed networks.

Finally, our work is also related to the work by Papadimitriou and Yannakakis [18], which deals with a setting in which the graph is specified dynamically. They seek dynamic decision rules that optimize the worst-case ratio of the distance covered to the length of the shortest path.

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

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

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 path P such that $\chi(P) \cdot \vec{w} \leq \chi(P') \cdot \vec{w}$ for every path P' in G.

3 Weight Discovery for DAGs: Upper Bounds

In this section we introduce two weight discovery protocols operating under the shared or routing cost models.

3.1 General DAGs

Consider some directed acyclic graph (DAG) G with a single source s and a single sink t.[1] Assume that G is *edge distinguishable*, namely, that 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 G is not edge distinguishable, namely, there exist two edges $e, e' \in E(G)$ such that for every path P in G, either $\{e, e'\} \subseteq P$ or $\{e, e'\} \cap P = \emptyset$, then $w(e)$ and $w(e')$ cannot be identified and G is not suitable for a weight discovery protocol. The edge distinguishability assumption can be lifted if one is interested in a shortest path discovery protocol, rather than a weight discovery protocol; implementing the former turns out to be a much easier task.

[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.

Outline. We establish the existence of a weight discovery protocol for G that operates in m rounds with $m+n-2$ agents (in total). This 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 3.1. *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 and 2 agents in the second round.*

Proof. In the first round send a single agent along the path P. In the second round send one agent along the path P and one agent along the path P'. The assertion follows trivially. □

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 following observation is a direct consequence of Observation 3.1.

Observation 3.2. *The vectors $\chi(Q)$, $\chi(P_{u,v})$, and $\chi(P_{s,u} \cup P_{v,t})$ can be spanned in 2 rounds with 3 agents (in total).*

We say that the simple path P is *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 3.3. *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 spanned.*

Proof. Let $P = (v_0, \ldots, v_k)$, where $s = v_0$ and $t = v_k$. To avoid cumbersome notation, we shall denote the vertex v_i by the integer i when this is clear from the context. The assertion holds

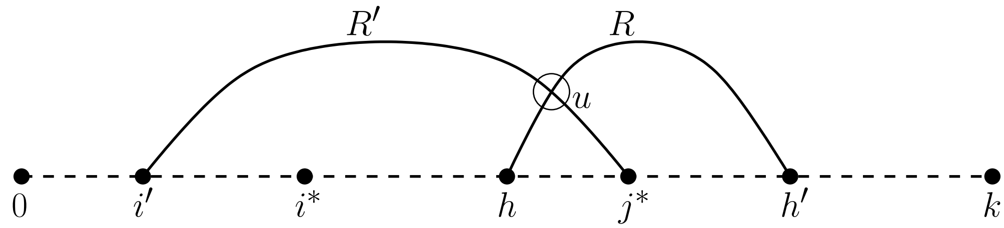

Figure 1: The path P (dashed line) and the bypasses R and R' (solid curves).

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: (1) $\chi(P_{i,j})$ can be spanned; (2) P admits an (i', j)-bypass for some $0 \leq i' \leq i$; and (3) P admits an (i, j')-bypass for some $j \leq j' \leq k$. This is well defined as Observation 3.2 guarantees that the start vertex and end vertex of the 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 spanned. Assume towards deriving 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 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 3.2 guarantees that $\chi(P_{h,j^*})$ can be 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 bypass R''.

So, assume that $R \cap R' = \emptyset$. By applying Observation 3.1 to the paths $P_{0,i'} \circ R' \circ P_{j^*,k}$ and $P_{0,h} \circ R \circ P_{h',k}$, we conclude that the characteristic vector of $P_{0,i'} \cup P_{h',k}$, which is the intersection of the two paths, can be spanned. This implies that $\chi(P_{i',h'}) = \chi(P) - \chi(P_{0,i'} \cup P_{h',k})$ can also be spanned. Since Observation 3.2 guarantees that $\chi(P_{h,h'})$ can be spanned (due to the bypass R) and that $\chi(P_{i',j^*})$ can be 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 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 3.3 can be employed to span the characteristic vectors of all edges in an edge distinguishable path.

Lemma 3.4. *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 $w(e)$ can be determined 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 3.3 guarantees that $w(e)$ can be determined for some edge $e = (u, v) \in P$. Let G' be the directed graph obtained from G by contracting the vertices u and v into a single vertex z. Let $P' = P - \{e\}$ be the path in G' which is obtained from P by contracting u and v into z. 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 under G' can be simulated by queries under G, thus we can apply Lemma 3.3 to G' and continue by induction[2] on $|P|$.

To that end, consider some (s,t)-path Q' in G'. If the edges of Q' form a path in G, then a query corresponding to Q' under G' is also a query under G. Otherwise, Q' must go through the vertex z. Moreover, the edge of Q' that enters z corresponds to an edge that enters u in G and the edge of Q' that exits z corresponds to an edge that exits v in G. Hence, there exists a path Q in G such that $Q = Q' \cup \{e\}$. Since $w(e)$ is 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. The fact that G is a DAG implies that every simple (s,t)-path in G is edge distinguishable. As every edge of G appears on some simple (s,t)-path in G, Lemma 3.4 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 3.3 and 3.4 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 theorem is a special case of Theorem 14.2.1 in [13].[3]

[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 [13] is somewhat different then 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

Theorem 3.5. *The dimension of the cycle space of a strongly connected digraph H is $|E(H)| - |V(H)| + 1$.*

By applying Theorem 3.5 to G' (which admits $n-1$ vertices and m edges), we conclude that the dimension of the (s,t)-path space in G is $m-n+2$. It follows that in $m-n+2$ rounds, each with a single agent, we can query a basis B for the (s,t)-path space of G.

A careful examination of the proofs of Lemmas 3.3 and 3.4 leads to the conclusion that they essentially rely on successive applications of the building block established in Observation 3.1. This building block consists of two rounds, the first with a single agent and the second with two agents. The basis B spans the queries corresponding to the single agent rounds, thus it can be extended to a basis for the whole edge space of G by appending to it $m-(m-n+2) = n-2$ vectors, each corresponding to a query involving two agents. The missing $n-2$ vectors are spanned in at most $n-2$ rounds, each with two agents. In total, a basis for the edge space of G is spanned in m rounds with $m+n-2$ agents.

3.2 DAGs in which All Vertices Have In-Degree ≥ 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. (Clearly, the same result holds if every internal vertex has out-degree at least 2.)

Theorem 3.6. *Let G be a DAG so that the in-degree of every internal vertex is at least 2. Then G admits a weight discovery protocol that operates in two rounds under the shared and routing cost models.*

Proof. 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 claim that with high probability, this protocol determines the weights of all edges.

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 3.7. *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.

coordinate. When the orientation of G induces a strongly connected directed graph, the flow space as defined in [13] coincides with the cycle space considered in the current paper.

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 $i, j \in \{1, 2\}$). $\square$

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. Combined with the fact that queries under G' can be simulated by queries under G (clearly, since the weights of the edges from s to v can be discovered, the answer of a query corresponding to an (s,t)-path in G can be translated to an (s',t)-path under G'.), we can apply Lemma 3.7 to G' and proceed by induction on the topological order to determine the weights of all edges in $E(G)$. The assertion of the theorem follows since the total probability of failure is bounded by $\frac{1}{2m} m \leq 1/2$. $\square$

Remark: The bound of 2 rounds is tight for this family of graphs, as demonstrated by the graph in Figure 2.

4 Shortest Path Discovery: a Lower Bound

Consider some undirected or directed acyclic graph G with two designated vertices $s, t \in V(G)$ so that every edge appears on some simple (s,t)-path of G. We show that any shortest path discovery

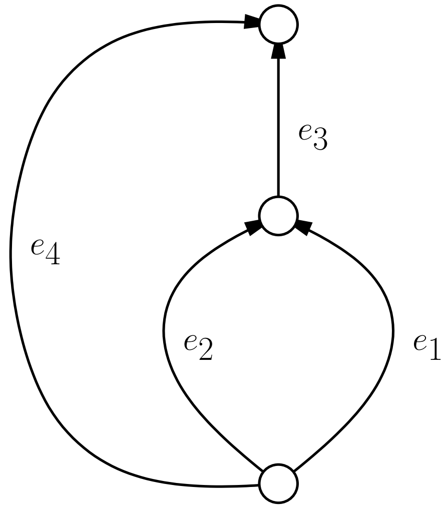

Figure 2: An example of a DAG that does not admit a shortest path protocol that operates in a single round under the shared and routing protocols.

protocol for G requires at least $m-n+1$ agents, regardless of the number of rounds and the cost model. We first establish the lower bound for directed acyclic graphs.

Theorem 4.1. *Any shortest path discovery protocol for a DAG 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 less 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 topological numbers of its incident vertices. 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 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 Section 3.1) and P is an (s,t)-path of G, the dimension of $\mathcal{C}$ is $m-n+1$. By the contradiction assumption, 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, there exists a sufficiently small $\epsilon > 0$, such that both $\vec{p} + \epsilon\vec{d}$ and $\vec{p} - \epsilon\vec{d}$ belong to the solution space of the attained system of equations (ϵ should be sufficiently small so that both $\vec{p} - \epsilon\vec{d}$ and $\vec{p} + \epsilon\vec{d}$ stay positive).

Consider the obtained solution $\vec{p} = \{p_e\}_{e\in E(G)}$. Let $\vec{d} \in \mathcal{C}$ and $\epsilon > 0$ be so that $\vec{p}+\epsilon\vec{d}$ and $\vec{p}-\epsilon\vec{d}$ are in the solution space, and let P' be some (s,t)-path satisfying

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

(which must exist as $\vec{d} \in \mathcal{C}, \vec{d} \neq 0$). To conclude the proof we show that the sign of $(\chi(P) - \chi(P'))\vec{w}$ cannot be determined.

Since all the queries have been answered as if all path lengths are equal, it follows that $\vec{p}(\chi(P) - \chi(P')) = 0$ for every path P'. 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 supposition that P has been discovered as a shortest path. $\square$

A close examination of the last proof implies 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 [15].

Lemma 4.2. *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) $C_1, \ldots, C_k$. 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 $1 \leq i \leq k-1$. Let $v_{i,i+1} \in V(G)$ denote the unique vertex that is shared by blocks C_i and C_{i+1} for $1 \leq i \leq k-1$. Let $s_i = v_{i-1,i}$ be the source of block C_i for every $2 \leq i \leq k$, and $t_i = v_{i,i+1}$ be the target of block C_i for every $1 \leq i \leq k-1$, so that $t_{i-1} = s_i$ for every $2 \leq i \leq k-1$. 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 [15], 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, $2 \leq i \leq k$, 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 $num(t_i) = |\bigcup_{j\leq i} V(C_j)|$. One can easily verify that the resulting graph is a DAG in which all (s, t)-paths have length n-1. $\square$

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 4.1 can be employed to establish the following corollary.

Corollary 4.3. *Any shortest path discovery protocol for an undirected graph requires at least $m - n + 1$ agents, independent 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.

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

Proposition 4.4. *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 less than two rounds.

Consider the routing cost model first. Clearly, the weight of the direct edge e_4 can be determined; suppose it is 2. Let x_1, x_2 denote the number of agents sent by the protocol along the paths P_1, P_2, respectively. The answers to the queries corresponding to P_1 and P_2 are $w_1x_1 + w_3(x_1 + x_2)$, and $w_2x_2 + w_3(x_1 + x_2)$. Suppose the answers are as if $w_1 = w_2 = w_3 = 1$ (so that all paths have the same length). Assume first that $x_1, x_2 \neq 0$. Let $\vec{d} = (-x_2(x_1 + x_2), -x_1(x_1 + x_2), x_1x_2)$. It is easy to verify that for a sufficiently small $\epsilon > 0$ the vectors $\vec{d}^+ = (1, 1, 1) + \epsilon\vec{d}$ and $\vec{d}^- = (1, 1, 1) - \epsilon\vec{d}$ are also valid solutions for the obtained system of equations. Now, consider the lengths of the paths

P_1, P_2, P_3 according to the two solutions. According to $\vec{d}^+$, their respective lengths are $2 - \epsilon x_2^2$, $2 - \epsilon x_1^2, 2$, so either P_1 or P_2 is a shortest-path. In contrast, according to $\vec{d}^-$, the respective lengths of P_1, P_2, P_3 are $2 + \epsilon x_2^2$, $2 + \epsilon x_1^2, 2$, 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. It remains to show that if either x_1 or x_2 equals zero, a shortest path cannot be discovered. Suppose $x_2 = 0$. In this case we have a single answer corresponding to $x_1(w_1 + w_3)$. It is easy to verify that, e.g., $(2, 0, 0)$ and $(0, 1, 2)$ are valid solutions. According to the former, P_2 is a shortest path, while according to the latter, P_1 and P_3 are shortest.

A similar analysis shows that a shortest path in G cannot be discovered within a single round under the shared cost model either. One can easily verify that if the answers are as if $w_1 = w_2 = w_3 = 1$ and $w_4 = 2$, two additional valid solutions are $(1, 1, 1) \pm \epsilon(-x_1, -x_2, x_1 + x_2)$ for some sufficiently small $\epsilon > 0$, and the lengths of P_1, P_2 are $2 \pm \epsilon x_2, 2 \pm \epsilon x_1$, respectively, while the length of P_3 is 2. Again, it is impossible to tell whether or not P_3 is a shortest path. $\square$

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 $x_2 = y x_1$ 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.

By Theorem 3.5, 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.

Proposition 4.5. *Every DAG admits a shortest path discovery protocol that operates in $m - n + 2$ rounds with a single agent in each round.*

The last proposition establishes the infeasibility of weight discovery protocols in DAGs under the independent cost model.

Proposition 4.6. *There exists no weight discovery protocol for DAGs 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 a non-direct edge (that has at least one incident internal vertex) can never be spanned. $\square$

5 Weight Discovery for Undirected Graphs: an Upper Bound

Consider an undirected graph G with two designated vertices $s, t \in V(G)$ so that every edge appears on some simple (s,t)-path. We establish the existence of a weight discovery protocol for G that operates in a single round with m agents.

Theorem 5.1. *Every undirected graph admits a weight discovery protocol that operates in a single round with at most m agents.*

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 costs 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 $x_i > c_i$ for every i, the matrix M is (strictly) diagonally dominant and therefore non-singular. 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 costs models the obtained matrix is diagonally dominant if $x_i > c_i$ for every i. Specifically, under the shared cost model, for every column i we have $M_{i,i} = x_i/(x_i+c_i)$, and for every $j \neq i$, $M_{j,i} \in \{1/(x_i+c_i), 0\}$, with exactly c_i non-zero entries. Similarly, under the routing cost model, for every column i, we have $M_{i,i} = x_i(x_i+c_i)$, and for every $j \neq i$, $M_{j,i} \in \{x_i + c_i, 0\}$, with exactly c_i non-zero entries. Applying the fact that $c_i \leq m-1$ for every i implies that setting $x_i = m$ for an odd m or $x_i = m+1$ for an even m results in a diagonally dominant matrix. □

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 claim that if the protocol chooses x_i randomly and uniformly to be either 1 or 3 for every i, the weights of all edges can be determined. In doing so, the paths taken by all agents are "almost" simple.

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., [2], Lemma 2.1). It follows that M is non-singular thus 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 $\vec{w}'$

References

[1] M. Aigner, *Combinatorial Search*, John Wiley and Sons, Chichester (1988).

[2] N. Alon, *Combinatorial Nullstellensatz*, Combinatorics, Probability and Computing 8 (1999), 7–29.

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

[4] 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.

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

[6] Itai Ashlagi, Dov Monderer, and Moshe Tennenholtz. Learning equilibrium in resource selection games. In *AAAI*, pages 18–23, 2007.

[7] 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)

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

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

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

[11] D. Fudenberg and D. Levine. *The theory of learning in games.* MIT Press, 1998.

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

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

[14] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of AI Research*, 4:237–285, 1996.

[15] 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.

[16] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. 11th ICML*, pages 157–163, 1994.

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

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

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

[20] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[21] M. Tan. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Proceedings of the 10th International Conference on Machine Learning*, 1993.

[22] M. Tennenholtz and Aviv Zohar. Learning equilibria in repeated congestion games. In *AAMAS*, pages 233–240, 2009.