Frugal Path Mechanisms

Aaron Archer*   Éva Tardos†

Abstract

We consider the problem of selecting a low cost $s-t$ path in a graph, where the edge costs are a secret known only to the various economic agents who own them. To solve this problem, Nisan and Ronen applied the celebrated Vickrey-Clarke-Groves (VCG) mechanism, which pays a premium to induce the edges to reveal their costs truthfully. We observe that this premium can be unacceptably high. There are simple instances where the mechanism pays $\Theta(n)$ times the actual cost of the path, even if there is an alternate path available that costs only $(1+\epsilon)$ times as much. This inspires the *frugal path* problem, which is to design a mechanism that selects a path and induces truthful cost revelation without paying such a high premium.

This paper contributes negative results on the frugal path problem. On two large classes of graphs, including ones having three node-disjoint $s-t$ paths, we prove that *no* reasonable mechanism can always avoid paying a high premium to induce truthtelling. In particular, we introduce a general class of *min function* mechanisms, and show that all min function mechanisms can be forced to overpay just as badly as VCG. On the other hand, we prove that (on two large classes of graphs) *every* truthful mechanism satisfying some reasonable properties is a min function mechanism.

1 Introduction

Many protocols in computer science involve the cooperation of multiple participants. For instance, interactions on the Internet typically involve many different components (servers, routers, etc.), controlled by different parties. If we assume that each participant, or *agent*, is selfishly motivated, we cannot expect that the agents will necessarily follow a given protocol, particularly if it is in an agent's self-interest to cheat. The field of *mechanism design*, which grew out of economic social choice theory, studies the design of protocols in which it is never in an agent's self-interest to cheat. These protocols resist manipulation because selfish agents behaving rationally would always choose to follow the rules.

A *mechanism* is a protocol that aggregates the privately known preferences of a group of agents into some *outcome* or "social choice." An agent might misrepresent her preferences in order to manipulate the mechanism. If it is never in a single agent's interest to do so, then the mechanism is called *truthful*, or *strategyproof*. If it is also never in the interest of a coalition of agents to jointly misrepresent their preferences, then the mechanism is called *group strategyproof*. The main tool used for eliciting truthful responses is to pay money to each agent for her participation, where the amounts depend both on the outcome chosen and on the agents' stated preferences.

*School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853. Email: aarcher@orie.cornell.edu. Supported by the Fannie and John Hertz Foundation. Part of this paper was written while visiting the IBM Almaden Research Center.

†Computer Science Department, Cornell University, Ithaca, NY 14853. Email: eva@cs.cornell.edu. Research supported in part by NSF grant CCR-9700163 and ONR grant N00014-98-1-0589.

In this paper we are interested in mechanisms whose payments are small by some measure. We describe these qualitatively as *frugal*. Often, mechanism design views the payments only as an inducement to the agents to bid truthfully, while the mechanism cares only about implementing some unrelated objective function (such as a social choice function, or finding the shortest path in a graph, or solving some machine scheduling problem). Of course, the literature on auctions does care about the payments, because the main point of selling something at auction is to make money.[1] Conversely, when the mechanism is paying to perform a task, it makes sense to try to keep the total payment low.

We consider the setting where the mechanism must accomplish some task, and it will hire a team of agents to do it. Each agent, if selected, performs a fixed service and incurs a fixed cost for performing that service. There exist certain teams of agents who can combine to perform the task, and no team is a subset of another. One special case is where the agents are edges in a network, we wish to send something through the network between nodes s and t, and the teams are $s - t$ paths. The catch is, an agent's cost is a secret known only to that agent. One can apply the celebrated VCG mechanism [25, 5, 10] to select the team that can perform the task while incurring the minimum cost *to itself*. However, the total amount that the *mechanism must pay* to the members of this team can be very high. In the network example, the cheapest team is just the shortest path with respect to the edge costs. Nisan and Ronen [20] indeed applied the VCG mechanism to solve this shortest path problem.

One drawback to the VCG mechanism when applied to the shortest path problem is that it can be forced to pay a huge amount, compared not only to the actual cost of the path, but also compared to the cost of the cheapest path that is completely disjoint from it. Roughly, we show that all reasonable path selection mechanisms can be forced to overpay just as badly as VCG. Thus, our results are negative. While we phrase everything in terms of paths because of their familiarity, our results also hold for the more general problem of hiring a team.

1.1 Vickrey auctions and VCG mechanisms

Historically, the first important mechanism is the Vickrey auction [25], or *second-price* auction. The basic version is a sealed-bid auction in which one item is sold to the highest bidder, but the price is equal to the second-highest bid. It is easy to see that, in the absence of collusion among agents (bidders), each agent's best strategy is to bid her true value, i.e., the highest price she would be willing to pay for the item. Thus, the mechanism is truthful. Of course, the Vickrey auction can be turned around, so that the agents are contractors bidding on a job. In this case, the auctioneer will hire the lowest bidder to do the job, and pay her the second lowest bid. Whenever the auctioneer acts as buyer, we will refer to the auction as a procurement auction.

The VCG mechanism addresses the more general problem of selecting from an arbitrary set of outcomes. In a VCG mechanism, each agent declares a real *valuation* function, which represents her preferences over the possible outcomes. The mechanism then selects the outcome that maximizes the total valuation of all agents, and computes a payment to each agent (which could be negative, if the agent is buying something). Assuming that each agent is trying to maximize the sum of her valuation and payment, the VCG mechanism is truthful (but not group strategyproof). Essentially, the VCG mechanism induces truthtelling by paying such that each agent's profit is equal to the value she adds to the system.

As discussed above, Nisan and Ronen [18] applied the VCG mechanism to network routing, in particular, finding shortest paths in a graph. If the mechanism chooses a path through edge e, this

[1] In this case, the payments are negative, because the agents are buying something from the mechanism.

edge incurs the (privately known) cost c_e (i.e., its valuation is $-c_e$). The VCG mechanism asks the edges to report their costs and chooses the shortest path according to the reported costs. Viewing this as a type of auction, we will refer to the reported costs as *bids*, and use b_e to denote edge e's bid. If there exists a cut edge e separating s from t, then the chosen path would have to use e no matter what, so no mechanism can elicit truthtelling by that edge. Thus, we assume that there is no cut edge, and hence there exist two edge-disjoint paths from s to t. The mechanism's payments are such that edge e's profit is exactly the value e adds to the system, i.e., the amount by which e's presence decreases the length of the shortest path. The mechanism pays zero to all edges not on the chosen path. To an edge e on the chosen path, it pays $\text{cost}|_{b_e=\infty} - \text{cost}|_{b_e=0}$. Here $\text{cost}|_{b_e=0}$ denotes the cost of the shortest path if e had bid zero, and $\text{cost}|_{b_e=\infty}$ is the cost of the shortest path in the graph with e deleted. Viewed differently, this payment is the highest value e could have bid and still been part of the shortest path. If e bids truthfully, then its profit is $\text{cost}|_{b_e=\infty} - \text{cost}|_{b_e=c_e}$, its added value. In the case that the graph consists of just some parallel arcs from s to t, the VCG mechanism for shortest paths reduces to the Vickrey auction.

1.2 How should we measure frugality?

The Vickrey auction has a nice property that we would hope to carry over to the shortest path problem. If we are buying a good or service from the lowest bidder, we would like to know that we are not losing too much by paying the second lowest bid. Of course, if the lowest bid is 1 and the second lowest is 100, then we are paying a big premium to enforce truthfulness. Since there are other truthful auction mechanisms, we might hope to find one that never pays more than some constant times the lowest bid. Unfortunately, elementary arguments show that this is impossible. Since Vickrey pays the second lowest bid, if there is tight competition in the market then the premium for truthfulness is probably not high, because the two lowest bids are likely to be close together. This suggests that we should consider a mechanism to be frugal if the total payment it makes is comparable to the cost of the second cheapest option. By this standard, the ordinary Vickrey procurement auction is frugal.

In contrast, the VCG shortest path mechanism can pay many times the cost of the chosen path, even if there is a disjoint alternate path of similar cost. Consider a graph consisting of two disjoint $s-t$ paths, P and Q. Suppose the length of path P (according to the bids) is L, the length of Q is $L(1+\epsilon)$, and P has k edges. Then the VCG mechanism chooses the path P. For each edge $e \in P$, the payment is $b_e + \epsilon L$, since this is the highest it can bid before Q becomes the shorter path. Thus, the total payment made to P is $L(1+k\epsilon)$. If $k\epsilon$ is large, then we could pay a large factor times the actual cost of the chosen path. Moreover, for fixed ϵ, we pay $\Theta(k)$ times the cost of the *longer* path. Thus, the VCG mechanism applied to the shortest path problem is not frugal.

The problem here is that the removal of any single edge of P forces us to take path Q. So, given the presence of the other edges, the marginal "added value" of an edge $e \in P$ is ϵL, even though it actually requires all of these edges to reap this benefit. If this entire path were one entity, then the VCG mechanism would pay it a profit of only ϵL, but since it is composed of k entities each of whom must be induced to bid truthfully, the mechanism must pay a total profit of $k\epsilon L$.

2 Our results

We show that no "reasonable" truthful mechanism avoids this pitfall. First we note that, roughly, *the payment scheme is entirely determined by the path selection rule* – the payment on an edge e in the chosen path should equal the *threshold* bid beyond which e would no longer be on the chosen path. Thus, if we restrict ourselves to selecting the shortest path, then VCG is the *only* truthful mechanism. However, it is conceivable that choosing an almost-shortest path might yield a better payment. Since each edge on the chosen path is paid at least its bid, we will still want to choose a fairly short path (with respect to the bids). The VCG mechanism paid a lot because it had to pay profits to many different edges. This suggests that we should bias our mechanism to give preference to paths with fewer edges.

A natural class of selection rules that allows us to easily introduce such a bias is what we call *min function* mechanisms. For each path, we evaluate some function of the bids on its edges, and select the path with the minimum function value. The VCG mechanism is a special case, where the function is just the sum of the bids. Unfortunately, an example similar to the bad example for VCG shows that min function mechanisms can be made to perform similarly poorly (see Section 7).

Since min function mechanisms exhibit essentially the same bad behavior as VCG, it is logical to look for other mechanisms. In Section 5 we outline four basic natural properties that we argue any reasonable path selection rule should have, and in Section 6 we verify that min function mechanisms satisfy them. We then prove a characterization result that on two reasonably large classes of graphs, every mechanism satisfying our reasonable properties is indeed a min function mechanism. One corollary of our result is that on any graph containing three node-disjoint $s-t$ paths, every mechanism satisfying our properties can be forced to exhibit the same bad behavior as the VCG mechanism. The two classes of graphs for which we prove our result are: (a) graphs with an $s-t$ arc, and (b) connected graphs containing s and t, with the addition of two node-disjoint $s-t$ paths.

We make some observations on the limits to which our characterization result might be extended. The two obvious questions are what happens on graphs outside the classes addressed by our theorems, and what happens if we relax some of the four desired properties of our mechanisms. We conjecture that our result holds for all graphs with three node-disjoint $s-t$ paths. However, if the graph consists of just two node-disjoint $s-t$ paths, then the characterization result fails. In Section 8 we provide an example of a truthful mechanism on these graphs which satisfies the desired properties of Section 5 but is not a min function mechanism.

Two of our four desired properties, while reasonable, seem technical and it seems plausible that one might be able to prove the characterization result without these assumptions (using an appropriately modified definition of min function mechanisms). However, two of them, the edge autonomy and independence properties, really are essential features of min function mechanisms. In Section 9 we show examples of truthful mechanisms that fail these two properties, thereby demonstrating that we really do need these assumptions.

Since our main results are negative, we generally do not concern ourselves with computational efficiency. However, in Section 10 we highlight one class of min function mechanisms that might be more frugal than VCG in practice, and show how to compute it in polynomial time.

3 The team and path selection problems

In this section, we define the general team selection problem and associated terminology, and specialize it to the path selection problem. We also state a well-known result about the structure of truthful mechanisms for these problems.

We define the general team selection mechanism design problem as follows. Let $\mathcal{N}$ denote the set of agents. Each agent is capable of performing a fixed service. Agent i incurs a cost of c_i for performing that service and zero otherwise. The cost c_i is secret data known only to agent i. Let $\mathcal{F} \subseteq 2^{\mathcal{N}}$ be a collection of the *feasible* subsets of agents. If $S \in \mathcal{F}$, this means that the team of agents S is capable of performing the desired task. The mechanism must select some feasible set of agents. Each agent i submits a bid b_i which is supposed to represent her true cost c_i, but it might be a lie. Let b denote the vector of bids. Based on these bids, the mechanism selects a team S, and determines a payment for each agent. We refer to the team S selected by the mechanism as the *winning* team, and say that agents $i \in S$ on this team *win*, while agents not on the team *lose*. Technically, a mechanism could make a non-zero payment to a losing agent, but our mechanisms will never do so.

A mechanism satisfies the *voluntary participation* property if every losing agent is paid a non-negative amount, and every winning agent is paid at least as much as her bid. We require our mechanisms to satisfy the voluntary participation property, since otherwise the question of minimizing payments doesn't make sense.

We assume that each agent's goal is to maximize her *profit*, which is the payment she receives minus the cost she incurs. A mechanism is *truthful* if, for each agent i, no matter what bids are submitted by the other agents, agent i's profit is maximized when she bids c_i. In other words, a mechanism is truthful if truthtelling is a dominant strategy for each agent. To facilitate discussion of this type of reasoning, we will use b_{-i} to denote the vector of bids by all agents besides i.

It is well known that a mechanism is truthful only if it has the "threshold property." This means that for each agent i and each fixed vector of bids b_{-i}, there exists a threshold bid $T_i(b_{-i}) \geq 0$ for agent i such that if i bids below $T_i(b_{-i})$ she will win, and if i bids above $T_i(b_{-i})$ she will lose. Thus, we must require that there is no agent i who belongs to every feasible team, or else there would be no threshold bid for i. It is easy to see that in a truthful mechanism, the payment to agent i should depend only on b_{-i} and whether i wins or loses. Thus, for fixed b_{-i}, there are only two possible payments that i can receive, and it is easy to show that the payment for winning must exceed the payment for losing by exactly $T_i(b_{-i})$. Since we are looking for truthful mechanisms that minimize payments subject to satisfying voluntary participation, we can assume that a winning agent i receives payment $T_i(b_{-i})$, whereas a losing agent receives zero payment. Therefore, there is no flexibility in our payment scheme. *The payment scheme is entirely determined by the rule we use for selecting the winning team.*

For the path selection problem, the agents are edges in the graph, there is a source s and a sink t, and the feasible sets are all possible paths from s to t. The edges can be directed or undirected. By convention, edges in a graph are traditionally denoted by the letter e, so we will tend to refer to the agents as e rather than i. Also, b_P will denote the vector of bids along a path (or any other set of edges) P, and b_{-P} the vector of all bids besides b_P. If e is a cut edge separating s and t, then it is on every feasible path. Thus, we assume that there is no cut edge, and hence there exist two edge-disjoint paths from s to t.

By the comments above, our path selection mechanisms are entirely determined by our rule for selecting the winning path. To enforce truthfulness, we need a path selection rule that is *monotone*, meaning that a losing edge can never cause itself to win by raising its bid, i.e., it induces threshold bids for the agents. The payment to a winning edge e will then be the largest amount it could have bid and still won.

Note that the edges are allowed to bid any positive real number, but not zero or ∞. However, we will come across cases where a bid is so large that it is "effectively" infinite, and we will use "bidding ∞" as a convenient shorthand for this. We will make this concept precise in Section 5.

4 Related Work

Recently there has been a spate of research at the border of economics, game theory and computer science. See [22] for a nice overview. The economics and game theory literature contains an enormous body of work on mechanism design, also called *implementation theory* or the *theory of incentives*. See [9, chapter 7], [17, chapter 23] or [21, chapter 10] for an introduction to the field, or the surveys [15, 11]. The celebrated Vickrey-Clarke-Groves (VCG) mechanism [25, 5, 10] is the main general positive result here. Nisan and Ronen [20] applied the mechanism design framework to some standard optimization problems in computer science, such as shortest paths and scheduling on unrelated machines, and raised some associated computational issues. The papers [1, 4, 12, 19] address computational difficulties with the VCG mechanism. The topic of revenue maximizing auctions has received a huge amount of attention. See [14] for a recent text on auction theory, or [6] for a survey on combinatorial auctions.

Feigenbaum et al. [8] performed some experiments to determine how much the VCG mechanism overpays on some artificial examples generated from the problem of routing between autonomous systems on the Internet. We will discuss their results in Section 12.

The issue of frugality was also raised by Archer and Tardos in [2], which presents a truthful mechanism for load balancing on machines. Their mechanism gives a constant factor approximation for this NP-hard problem and pays the machines only a logarithmic factor more than the actual costs incurred by the machines, provided no one machine dominates the total processing power. The behavior of the VCG mechanism applied to the minimum spanning tree (MST) problem is considered in Bikhchandani et al. [4]. They prove that if T_1 is the MST and T_2 is the MST in the remaining graph after T_1 has been removed, then the VCG mechanism does not pay more than $\text{cost}(T_2)$. Both of these results contrast starkly with the main result of this paper that, under suitable assumptions, truthful path selection mechanisms can be forced to pay $\Theta(n)$ times the actual costs incurred, even in a tight market.

Since the original appearance of this paper [3], several authors have followed up on our work. Elkind et al. [7] proved that every truthful mechanism (without any additional assumptions) can be forced to overpay just as badly as VCG in the worst case. That is, they were able to extend our non-frugality result (our Theorem 7.1) to all graphs, and without assuming the mechanism has the desired properties we specify in Section 5. They also examine mechanisms where they assume a commonly known probability distribution on edge costs, and they wish to find a mechanism for which truth-telling is a Bayesian Nash equilibrium. This is a much weaker solution concept than that of dominant strategies (as we consider), and hence it admits a larger class of mechanisms. Remarkably, they show that the mechanism in this class that minimizes the total expected payments (where the expectation is over the probability distributions on the edges) *is* a particular type of min function mechanism. In other words, even though they are considering a wider class of admissible mechanisms, the optimal mechanism in this class is still truthful according to the stronger definition used in our work. This optimal mechanism is a min function mechanism that is tuned for the given probability distributions. They also give examples of distributions where the expected total cost (to the agents) of the shortest path is $\Theta(n)$, as is the expected total payment by the optimal min function mechanism, but the VCG mechanism pays $\Theta(n\sqrt{n})$ in expectation. This gives further justification for the study of min function mechanisms.

Whereas our work examines one interesting mechanism design setting where VCG overpays badly and shows this drawback is inherent to all truthful mechanisms for the problem, Talwar [24] takes an alternate approach of trying to characterize for what problems the VCG mechanism is guaranteed not to overpay by much. In particular, Talwar examines the general team selection problem and considers imposing structure on the collection of feasible sets. Recall the result of Bikhchandani et al. [4] that when the collection of feasible sets $\mathcal{F}$ are all of the spanning trees of a graph, the VCG mechanism never pays more than the cost of the cheapest feasible set that is disjoint from the one chosen. This is related to the fact that the collection of forests in a graph forms a matroid. Talwar's main result is that applying the VCG mechanism to a team selection problem yields this worst-case guarantee if and only if the collection of feasible sets $\mathcal{F}$ is what he calls a "frugoid," which is a generalization of a matroid.

For combinatorial auctions, Lavi et al. [16] prove a characterization result similar in spirit to our Theorems 6.5 and 6.4. Under conditions similar to ours, they prove that a slight generalization of VCG is the only class of truthful mechanisms for combinatorial auctions. Their characterization theorem is reminiscent of Roberts's classic result [23] that in a general mechanism design problem where the agents' valuations of the outcomes can be arbitrary, a truthful mechanism must select the outcome that maximizes an affine function of the valuations. However, the methods they use are very different from Roberts's, and they credit our work for several key ideas in their proof.

5 Desired properties for truthful mechanisms

We discuss some properties that we argue any reasonable path selection rule ought to possess. First, each edge and path should be able to control its own fate in a limited way. In an ordinary auction (with the agents selling), it is a standard requirement that any agent bidding high enough will lose and low enough will win, provided the other agents' bids are fixed. In the path selection setting, it could be that every path that contains a particular edge e contains other very expensive edges, so edge e would lose no matter how low she bids. However, if all of the edges along a path P bid sufficiently low, then path P should win. One way to look at this is that if the total cost of path P is sufficiently low compared to *every* other path (since every other path contains at least one edge not in P), then P should win. On the other hand, if there is at least one cheap path P, then any edge $e \notin P$ should be able to force itself to lose by bidding sufficiently high. Next, since we are comparing the merits of various paths, it makes sense that raising the bids of edges not on path P should only make P more attractive. In particular, if P is the winner, it should still win even if other edges bid higher. We call this the *independence* property, because it implies that our comparison between paths P and Q is independent of the bids on all other edges. We summarize these properties below.

- (path autonomy) Given any bids b_{-P} for the edges off the path P, if all the edges on P bid sufficiently low then P will win.
- (edge autonomy) Suppose P is a path, edge $e \notin P$, and the bids along P are fixed at b_P. Then there exists some bid T such that if e bids T or higher, no path using e will win, no matter what the bids of the edges not in $P \cup \{e\}$.
- (independence) If path P wins, and an edge $e \notin P$ raises its bid, then P will still win.

Our last property, which we call *sensitivity*, is the most technical. It essentially says that if two paths are "tied," then perturbing the bids appropriately should break the tie.

Definition 5.1 *Suppose that for some set of bids, path P wins. Suppose that there exists an edge e such that arbitrarily small perturbations of e's bid cause another path Q to win. Then we say paths P and Q are* tied for the lead.[2]

[2]Note that, by independence, e belongs to either P or Q.

Sensitivity just says that the mechanism should pay attention to all of the bids. If P wins but is tied with Q for the lead, then any increase in the bid of any edge in $P-Q$ or any decrease in the bid of any edge in $Q-P$ should make Q more attractive than P. Thus, any of these perturbations should make P lose. However, they may not make Q win, since there may be other paths also tied for the lead. The rigorous definition appears below.

- (sensitivity) Suppose path P wins, and Q is tied with P for the lead. Then increasing b_e for any $e \in P-Q$ or decreasing b_e for any $e \in Q-P$ causes P to lose.

We now record a simple consequence of the independence property to which we have already alluded.

Proposition 5.2 *If lowering the bid on some edge e causes the winning path to change, then the new winning path must contain e.*

Proof: Suppose P wins, but lowering the bid on edge e causes path Q to win, where $e \notin Q$. Raising e's bid back to its original value changes the winning path back to P, contradicting independence. ■

Bidding "infinity." Now that we have defined independence and edge autonomy, we can state precisely what we mean when we say that an edge e "bids ∞." We mean that we are about to consider several settings for the bids along particular paths in the graph that do not use edge e. For each of these settings, the edge autonomy property guarantees there is a threshold such that if b_e is above the threshold, e loses. Thus, if we set b_e anywhere above the maximum of these thresholds, e loses for each of these settings. By the independence property, the precise value of b_e does not matter, as it does not affect which path wins. To avoid running through this argument every time, we simply write that "e bids ∞."

6 Min function mechanisms

We define a broad class of truthful mechanisms called min function mechanisms, that allow us to easily introduce a bias toward certain paths, say ones with fewer edges. For each $s-t$ path P in the graph, define a positive real valued function f_P of the vector of bids b_P. The mechanism evaluates each function $f_P(b_P)$ and selects the path with the lowest function value, breaking ties using any arbitrary rule that does not depend on the bids of edges not involved in the tie. One example of a valid tiebreaking rule would be to establish a total order on the paths (not depending at all on the bids) and break ties according to this order. (Since there are possibly exponentially many paths, we cannot necessarily compute these mechanisms in polynomial time, even if the functions are simple. But this is of no concern to us at present, since we are after a characterization result.) In order for the mechanism to be truthful and satisfy the desired properties, we require the functions to satisfy the following:

- $f_P(b_P)$ is continuous and strictly increasing in b_e, for each $e \in P$
- (infinite limit) $\lim_{b_e \to \infty} f_P(b_P) = \infty$ for each $e \in P$
- (zero limit) $\lim_{b_P \to 0} f_P(b_P) = 0$.

Notice that the VCG shortest path mechanism is a special case, where the function f_P simply sums the bids along the path P: $f_P(b_P) = \sum_{e \in P} b_e$. Another example of a min function mechanism is to set $f_P(b_P) = \ln(1 + |P|) \sum_{e \in P} b_e$. This would introduce a mild bias against paths with many edges. Another example would be $f_P(b_P) = \prod_{e \in P} b_e^2$, although it is not clear why this one would be useful.

Theorem 6.1 *The min function path selection rule yields a truthful mechanism.*

Proof: We must show that this path selection rule is monotone, i.e., a losing edge can never cause itself to win by raising its bid. If P is currently the winning path and $e \notin P$, then $f_P(b_P)$ is the minimum function value. Raising b_e strictly increases the function values for all paths Q containing e, but does not affect f_P. Thus, Q cannot win because P beats it. So the mechanism is monotone.

We now show that every edge in the winning path has a threshold bid. Let P be the path not including e whose function value is minimum. (It exists because the graph is 2-edge connected.) Let T be the largest bid for e such that some path Q containing e has $f_Q(b_{Q-e}, T) \leq f_P(b_P)$. As e raises its bid towards T, the winning path could change, but always contains e. Once b_e is raised beyond T, path P wins. ■

Notice that our proof relies only on the functions f_P increasing with infinite limit. Continuity and the zero limit are not important for truthfulness, and we can dispense with the strictness if we employ an appropriate tie breaking rule (such as numbering the paths and breaking ties by the lowest numbered path). We included these extra requirements on the f_P's so that min function mechanisms will satisfy our desired properties.

Theorem 6.2 *Min function mechanisms satisfy the edge and path autonomy, independence and sensitivity properties.*

Proof: Path autonomy follows from the zero limit property, since the functions take strictly positive values. Consider distinct paths P and Q. As we lower the bids b_P of edges in P to zero, $f_Q(b_Q)$ may decrease (if P and Q share some edges). However, since Q contains some edge not in P, $f_Q(b_Q)$ remains bounded away from 0. Thus, eventually $f_P(b_P)$ dips below $f_Q(b_Q)$. Thus, for b_P sufficiently close to the zero vector, path P wins.

Edge autonomy follows from the fact that the functions are increasing with infinite limits. Suppose P is a path, b_P is fixed, edge $e \notin P$, and e is on some path Q. Then for sufficiently high bids b_e we have $f_P(b_P) < f_Q(b_Q)$, so Q loses. Since e lies on a finite number of paths, a sufficiently high bid causes e to lose.

Independence follows because the f_P's are strictly increasing and unaffected by the edges not on P, and the tie-breaking does not depend on edges not involved in the tie. If path P wins, edge $e \notin P$, and e raises its bid, this cannot cause any path Q containing e to win, as we already argued in the proof of Theorem 6.1. If any other path R were to win, then P and R must have been tied. But the tiebreaking between paths P and R does not depend on e, since $e \notin P \cup R$. So P still wins.

Sensitivity follows because the f_P's are continuous and *strictly* increasing. By continuity, if P and Q are tied for the lead, we must have $f_P(b_P) = f_Q(b_Q)$. Since the functions are strictly increasing, raising the bid of any edge $e \in P - Q$ or lowering the bid of any $e \in Q - P$ causes Q to beat P.

■

6.1 Characterization results

Theorems 6.1 and 6.2 show that on any graph, a min function mechanism is truthful and satisfies the desired properties. We now show a partial converse, namely that on two large classes of graphs, every truthful mechanism satisfying the desired properties is a min function mechanism. The first class is all graphs that contain an $s-t$ arc. The second class is all graphs consisting of a connected graph containing s and t, plus two parallel $s-t$ paths whose nodes are disjoint from each other and from the rest of the graph.

We would like to have some concept of the bids on one path being "less than" the bids on another path. As we have noted, the independence property implies that, when deciding the winner between paths P and Q, the bids on the other edges are irrelevant. Thus, we might as well assume that the other edges all bid ∞. Suppose P bids b_P, Q bids b_Q, and all other edges bid ∞. If P wins, then we will write $b_P \prec b_Q$. For this to be defined, it is necessary that the bids b_P and b_Q can be *realized simultaneously*, i.e., they agree on the common edges $P \cap Q$. Note that if P and Q share a node (other than s and t) then we can put together pieces of P and Q to create another finite path L. If L is the winner, then we cannot necessarily compare b_P and b_Q. In this case, we say that b_P and b_Q are *incomparable*. If P and Q are node-disjoint then they are the only two finite paths, so one of them must win and the bids are comparable.

A small word may be in order here to justify and clarify the notation. In a min function mechanism, the function f_P is really $f_P : \mathbb{R}_+^P \to \mathbb{R}_+$. That is, the argument of $f_P(\cdot)$ should be a vector of positive reals, whose coordinates are indexed by the edges in path P. Similarly, when we write $x \prec y$, x and y must each be vectors of positive reals, whose coordinates are indexed by the edges of a path, i.e., we must have $x \in \mathbb{R}_+^P$ and $y \in \mathbb{R}_+^Q$ for some paths P and Q. If x is an explicit vector, e.g., the vector of all 1's, the notation is meaningless unless we specify which edges these bids correspond to. Thus, if x is an explicit vector that we intend to interpret as a vector of bids for path P, we will write it as $(x)_P$. For example, if $c \in \mathbb{R}_+$ then $(c \cdot \mathbf{1})_P$ denotes bids of c for each edge of P. If P consists of a single arc, then $(c)_P$ denotes a bid of c on that one arc.

Lemma 6.3 *Given a mechanism satisfying the independence property, suppose P, Q, and L are three $s-t$ paths, the bids b_P and b_L can be realized simultaneously and are comparable, and Q is node-disjoint from P and L. If $b_P \prec b_Q$ and $b_Q \prec b_L$, then $b_P \prec b_L$.*

Proof: Suppose on the contrary that when P bids b_P, L bids b_L, and the other edges all bid ∞, path L wins. Now lower the bids on Q to b_Q. By Proposition 5.2, either L still wins, or some path containing an edge in Q wins. But since Q is node-disjoint from P and L, every such path besides Q contains an edge that bid ∞. Thus, either L or Q wins. If L wins, then independence implies it still wins after we raise the bids on all edges in $P-L$ to ∞, but this contradicts the assumption $b_Q \prec b_L$. If Q wins, then it still wins after we raise all the bids on $L-P$ to ∞, which contradicts $b_P \prec b_Q$. ∎

Theorem 6.4 *If graph G contains the edge $s-t$, then any truthful mechanism for the $s-t$ path selection problem on G that satisfies the independence, sensitivity and edge and path autonomy properties is a min function mechanism.*

Proof: For each $s-t$ path P and each possible vector of bids b_P for the edges along path P, we must first define a value for $f_P(b_P)$. Let R denote the path consisting of the single edge $s-t$. We call this the *reference* arc. Since R is node-disjoint from every other $s-t$ path, $(c)_R$ is comparable to every bid for every other path. Define $f_R((c)_R) = c$. For each other path P and set of bids b_P, define

$$f_P(b_P) = \sup\{c : (c)_R \prec b_P\}. \tag{1}$$

In other words, $f_P(b_P)$ is defined to be the threshold bid for the reference arc when P bids b_P and all other edges bid ∞. Since the mechanism is truthful, R will win if it bids below this threshold, and lose if it bids above the threshold. By path autonomy, R will win if it bids sufficiently low. Thus, $f_P(b_P) > 0$.

Now we must show that the functions we have defined select the correct path. Suppose on the contrary that $f_P(b_P) < f_Q(b_Q)$, but the mechanism selects path Q. By independence, if we raise all other bids to ∞, then Q still wins, so $b_Q \prec b_P$. By definition of f_P and f_Q, neither P nor Q can be R. Now let $c \in (f_P(b_P), f_Q(b_Q))$, so $b_P \prec (c)_R$ and $(c)_R \prec b_Q$. Applying Lemma 6.3 yields $b_P \prec b_Q$, which is a contradiction.

Finally, we must prove that the functions f_P satisfy the three properties we require of the functions defining a min function mechanism. Edge autonomy implies that $f_P(b_P) \to \infty$ as any bid along P goes to ∞, and path autonomy implies that $f_P(b_P) \to 0$ as $b_P \to 0$, since no matter what the reference arc bids, P must be able to lose to or beat it. Truthfulness implies the functions f_P are monotone increasing, since raising b_e can never cause a path containing e to beat the reference arc. Strict monotonicity and continuity are obvious for f_R, but are a bit more involved for the other paths.

To show the f_P's are strictly increasing, we fix some path P, some $e \in P$, and fix b_P to some bid $\bar{b}_P$. Set R's bid to $f_P(\bar{b}_P)$, and all other bids to ∞. Suppose that R wins (a similar argument holds when P wins). Then any increase to b_R causes P to win. Thus any decrease to b_e causes P to win (by sensitivity). Let us decrease b_e by ϵ, so that $b_P = (\bar{b}_{P-e}, \bar{b}_e - \epsilon)$ and P wins. We want to show that the threshold bid for R is lower, which shows that $f_P(b_P)$ is lower. If not, than any decrease to R's bid causes R to win. So by sensitivity, any increase in b_e causes R to win. In particular, raising b_e by $\frac{\epsilon}{2}$ so that $b_P = (\bar{b}_{P-e}, \bar{b}_e - \frac{\epsilon}{2})$ causes R to win. But we already argued that P wins in this situation, so we've reached a contradiction.

To show continuity, we again argue by contradiction. Suppose we have some path P such that f_P is not continuous. Then f_P is discontinuous in some coordinate b_e, where e is an edge of P. Since f_P is monotone increasing in b_e, there must be some jump discontinuity at some point $b_P = \bar{b}_P$, where $f_P(\bar{b}_{P-e}, \bar{b}_e^-) = x$, $f_P(\bar{b}_{P-e}, \bar{b}_e^+) = y$, $x < y$, and the $^+$ and $^-$ denote limits as $b_e \to \bar{b}_e$ from above and below. Suppose $f_P(\bar{b}_P) = x$ (a similar argument holds when $f_P(\bar{b}_P) = y$). Suppose R bids $\frac{x+y}{2}$ and P bids $\bar{b}_P$, so P wins. Any increase in e's bid would raise $f_P(b_P)$ above y, so R would win. Thus, by sensitivity, any decrease in R's bid causes R to win, in particular for $b_R \in (x, \frac{x+y}{2})$, which is a contradiction. ∎

Theorem 6.5 *If the graph G consists of some connected graph including nodes s and t, plus two extra $s-t$ paths that are disjoint from the rest of graph, then any truthful mechanism for the $s-t$ path selection problem on G that satisfies the independence, sensitivity and edge and path autonomy properties is a min function mechanism.*

Proof: Let $\mathcal{M}$ denote the mechanism. Let us call one of the disjoint paths R (for *reference* path) and the other disjoint path S (for *secondary* reference path). Define $f_R((c \cdot \mathbf{1})_R) = c$, where $\mathbf{1}$ denotes the vector of all ones. That is, when every edge on path R bids c, we define the corresponding function value to be c. We also need to define $f_R(b_R)$ when the bids are not all the same, but let us postpone this for now. For any $P \neq R$, define

$$f_P(b_P) = \sup\{c : (c \cdot \mathbf{1})_R \prec b_P\}.$$

Notice that $b_P \prec (c \cdot \mathbf{1})_R$ for all $c > f_P(b_P)$, which follows directly from the definition of $f_P(b_P)$. Also, $(c \cdot \mathbf{1})_R \prec b_P$ for all $c < f_P(b_P)$ which follows from the definition and the fact that the path selection rule is monotone.

The proof that the functions f_P for $P \neq R$ are strictly increasing, continuous, and have the two limit properties, is exactly the same as in the proof of Theorem 6.4, except that every time we set the reference arc bid to c in that argument, here we set the vector of bids on the reference path to $(c \cdot \mathbf{1})_R$. Therefore, the range of each function f_P (for $P \neq R$) is all of $(0, \infty)$.

We want to show that the mechanism $\mathcal{M}$ always selects a path of minimum function value. In other words, if $f_P(b_P) < f_Q(b_Q)$, we need to rule out the possibility that Q wins. If Q were to win, then, by the independence property, it would still win after we raise the bids of all edges not in P or Q to ∞, so we would have $b_Q \prec b_P$. Thus, it suffices to rule out this possibility. In other words, we want to show that if $f_P(b_P) < f_Q(b_Q)$ and b_P and b_Q are comparable, then $b_P \prec b_Q$. Even though we have not yet defined the function f_R on its entire domain, we can already dispense with the case when neither P nor Q is R.

Case 1: $f_P(b_P) < f_Q(b_Q)$ and neither P nor Q is R. Select $c \in (f_P(b_P), f_Q(b_Q))$. Then by the observation above, we have $b_P \prec (c \cdot \mathbf{1})_R \prec b_Q$. Thus, we can apply Lemma 6.3 to obtain $b_P \prec b_Q$.

Next we must define $f_R(b_R)$ when the bids along R are not all the same. Let

$$f_R(b_R) = \sup\{f_S(b_S) : b_S \prec b_R\}.$$

Directly from this definition we see that $f_R(b_R) < f_S(b_S)$ implies that $b_R \prec b_S$. Now that we have shown the other functions f_P to have the properties we want, we can use them to prove the properties for f_R.

Case 2: $f_P(b_P) < f_R(b_R)$ and $P \neq S$. By definition of $f_R(b_R)$, there exist vectors of bids b_S such that $f_S(b_S)$ is arbitrarily close to but not greater than $f_R(b_R)$, while $b_S \prec b_R$. In particular, there exists $\bar{b}_S \in \mathbb{R}_+^S$ such that $\bar{b}_S \prec b_R$ and $f_S(\bar{b}_S) \in (f_P(b_P), f_R(b_R)]$. By Case 1 we have $b_P \prec \bar{b}_S$. Applying Lemma 6.3 gives $b_P \prec b_R$.

Case 3: $f_R(b_R) < f_P(b_P)$ and $P \neq S$. Since the range of f_S is all of $(0, \infty)$, there exists $\bar{b}_S \in \mathbb{R}_+^S$ such that $f_S(\bar{b}_S) \in (f_R(b_R), f_P(b_P))$. By definition of $f_R(b_R)$, we must have $b_R \prec \bar{b}_S$. By Case 1, we have $\bar{b}_S \prec b_P$. Applying Lemma 6.3 gives $b_R \prec b_P$.

Case 4: $f_S(b_S) < f_R(b_R)$. Select any other $s - t$ path P in the graph. Since the range of f_P is all of $(0, \infty)$, there exists a bid $\bar{b}_P$ such that $f_P(\bar{b}_P) \in (f_S(b_S), f_R(b_R))$. By Case 1 we have $b_S \prec \bar{b}_P$, and by Case 2 we have $\bar{b}_P \prec b_R$, so by Lemma 6.3 we have $b_S \prec b_R$.

Case 5: $f_R(b_R) < f_S(b_S)$. By definition of $f_R(b_R)$, we must have $b_R \prec b_S$.

For $P \neq R$, we already know that f_P is strictly increasing, continuous, and has the two limit properties. Now that we know the mechanism selects a path of smallest function value as the winner, we can prove that f_R also has these properties, using arguments similar to the proof of Theorem 6.4. Thus, our functions satisfy the required properties to define a min function mechanism, and $\mathcal{M}$ always selects a path of minimum function value. Since a min function mechanism may break ties arbitrarily, we choose ours to break ties the same way that $\mathcal{M}$ does. Thus, we have successfully represented $\mathcal{M}$ as a min function mechanism. ■

Extensions. Even though our original motivation for studying min function mechanisms was to show that non-frugality is inherent in the path selection problem, we think that the characterizations given by Theorems 6.4 and 6.5 are surprising and inherently interesting in their own right. Thus, even though Elkind et al. [7] have now proven the non-frugality result for all graphs without using min function mechanisms, it would still be interesting to know how far one can extend our characterization result. This would include both extending the class of graphs on which it holds, and determining which of the assumptions of Section 5 we can dispense with.

We conjecture that Theorem 6.5 extends to all graphs containing 3 node-disjoint $s-t$ paths P, Q, and R. In such a graph, when all other edges incident to every internal node of P or Q bid ∞, the mechanism reduces to a case covered by Theorem 6.5. It seems that one ought to be able to extend the min function representation to the whole graph. On the other hand, in Section 8 we show an example of a mechanism for the graph consisting of two disjoint $s-t$ paths that satisfies the desired properties, but is not a min function mechanism.

Since min function mechanisms satisfy edge and path autonomy, independence and sensitivity, clearly we cannot relax any of these conditions and still hope to prove that the mechanism is a min function mechanism. However, it may be possible to extend our characterization result by simultaneously relaxing our assumptions on the mechanism and our requirements of the functions f_P defining a min function mechanism. Which assumptions/requirements might we relax? The independence property is inherent in selecting paths based on comparing their function values. Truthfulness requires edges to have threshold bids, and monotonicity and infinite limits seem the most natural way to induce thresholds. The edge autonomy property follows directly from the infinite limits. Thus, independence and edge autonomy are essential properties of min function mechanisms. Requiring these properties is not a vacuous condition: in Section 9 we give two examples of truthful mechanisms where these properties fail.

We required sensitivity and path autonomy for technical convenience. Indeed, if we dispense with the requirements of zero limit, continuity, and *strict* increase and instead require only that the functions in a min function mechanism be increasing with infinite limits, we are still left with a truthful mechanism. (Over ranges where f_P is constant, we must be a bit careful with tiebreaking, but this can be done.) It seems plausible that one could prove that a mechanism satisfying just independence and edge autonomy must be a min function mechanism under this new definition.

7 Costly example for min function mechanisms

We show here that any min function mechanism can be forced to pay $L(1+k\epsilon)$, where L is the cost of and k is the number of edges on the winning path, even if there is some node-disjoint path of cost $L(1+\epsilon)$. If $k\epsilon$ is large, then we would pay a large factor times the actual cost of the chosen path, even though we are in a "tight market." Moreover, for fixed ϵ, we pay $\Theta(k)$ times the cost of the *longer* path. This contrasts with the Vickrey auction, where we only pay as much as the 2nd lowest bid.

Consider a min function mechanism on a graph consisting of two node-disjoint $s-t$ paths, P and Q. Fix some $L, \epsilon \in \mathbb{R}_+$, and let $|P|$ and $|Q|$ denote the number of edges in P and Q. Let b_P^i denote the vector of bids along P where all edges bid $\frac{L}{|P|}$, except for the ith edge, which bids $\frac{L}{|P|} + \epsilon L$. Similarly, let b_Q^i denote the bids along Q where all edges bid $\frac{L}{|Q|}$, except for the ith edge, which bids $\frac{L}{|Q|} + \epsilon L$. Suppose without loss of generality that $f_Q(b_Q^1)$ is the maximum of the function values $f_P(b_P^1), \ldots, f_P(b_P^{|P|}), f_Q(b_Q^1), \ldots, f_Q(b_Q^{|Q|})$. Let b_P^0 denote the vector of bids when every edge along P bids $\frac{L}{|P|}$. Now consider how the mechanism acts when P bids b_P^0 and Q bids b_Q^1. Since $f_P(b_P^0) < f_P(b_P^1) \le f_Q(b_Q^1)$, path P wins. To determine the payments, we must figure out the threshold bid for each edge in P. Since, for $i = 1, \ldots, |P|$, $f_P(b_P^i) \le f_Q(b_Q^1)$, P still wins (or ties) even if one of the edges along P raises its bid by ϵL. Thus, the threshold bid for each edge in P is at least $\frac{L}{|P|} + \epsilon L$, so the total amount paid by the mechanism is at least $L(1 + |P|\epsilon)$, even though the length of P is L and Q is only $L(1+\epsilon)$. If the graph consists of more than just P and Q, we can achieve the same result by setting the bids of all other edges to be ∞.

Theorem 7.1 *Any truthful mechanism on a graph that contains either an $s-t$ arc or three node-disjoint $s-t$ paths and satisfies the independence, sensitivity and edge and path autonomy properties can be forced to pay $L(1 + k\epsilon)$, where the winning path has k edges and costs L, even if there is some node-disjoint path of cost $L(1+\epsilon)$.*

Proof: If the graph G belongs to one of the two classes covered by Theorems 6.4 and 6.5, then the mechanism is a min function mechanism. Otherwise, select three node-disjoint paths, P, Q, and R. When all of the other edges of G bid ∞, we can essentially ignore them, by the independence property. We can then apply Theorem 6.5 to the remaining graph, which consists of P, Q, and R. Thus, the mechanism reduces in this case to a min function mechanism. The preceding discussion shows how any min function mechanism can be forced to pay the specified amount. ∎

8 An exceptional mechanism

In this section, we exhibit a truthful mechanism (for a graph outside our two classes) that satisfies the independence, sensitivity and edge and path autonomy properties but is not a min function mechanism. Let the graph G be a 4-node graph consisting of 2 $s-t$ paths P and Q, each having two edges. Let x_1 and x_2 be the bids on the two edges of P, and y_1 and y_2 be the bids on the two edges of Q. Let us describe the winning path via a "phase diagram." For a given (y_1, y_2), draw a dividing line in the $x_1 - x_2$ plane with x_1-intercept $(y_1 + \frac{y_2}{2}, 0)$ and x_2-intercept $(0, \frac{y_1}{2} + y_2)$. If (x_1, x_2) lies on or below the line, then P wins. Otherwise Q wins. In Figure 1, P loses if it bids x^1, wins if it bids x^2, and ties but wins the tiebreaker if it bids x^3. (Note that x^i is a vector of bids, $x^i = (x_1^i, x_2^i) \in \mathbb{R}_+^P$.)

It is easy to check that this mechanism is truthful and satisfies the desired properties. For P, the threshold bids are where the horizontal and vertical lines through (x_1, x_2) cross the dividing line in the phase diagram, provided this crossing occurs in the first quadrant, and zero otherwise. Increasing either bid along Q pushes the dividing line both up and to the right in the phase diagram, which cannot cause Q to win unless it was already the winner. Since increasing either bid can push the dividing line out arbitrarily far, the edges of Q have threshold bids as well. Thus, the mechanism is truthful. Since, for fixed (y_1, y_2), the region of the phase diagram where P wins contains a neighborhood of the origin, P can win by bidding low enough. Path Q can also win by bidding sufficiently low, since this moves the dividing line arbitrarily close to the origin. Hence path autonomy holds. With b_Q fixed, either edge on P can cause itself to lose by bidding larger than the appropriate intercept of the dividing line defined by b_Q. With b_P fixed, either edge on Q can cause itself to lose, because both intercepts of the dividing line go to ∞ as either edge's bid does. Thus, edge autonomy holds. Independence holds trivially because there are only two paths

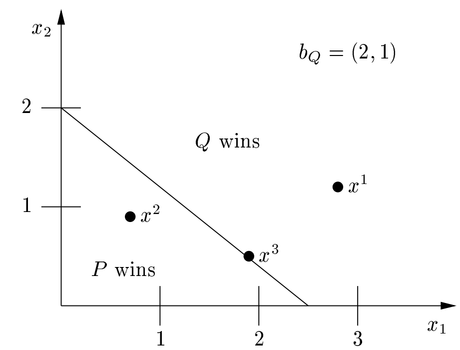

Figure 1: The winning path is determined by this “phase diagram.”

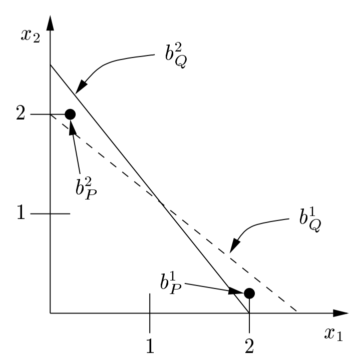

Figure 2: This figure shows that the $\prec$ relation on bids is non-transitive.

from which to choose. The two paths are tied for the lead exactly when (x_1, x_2) is on the dividing line. In this case P wins, but increasing either bid on P causes the point to move above the line so Q wins, and decreasing either bid on Q causes the line to move down and left, so Q wins. Thus, sensitivity holds.

To show that this mechanism cannot be represented by a min function mechanism, consider the bids $b_P^1 = (2, \frac{1}{5})$, $b_P^2 = (\frac{1}{5}, 2)$, $b_Q^1 = (2, 1)$ and $b_Q^2 = (1, 2)$. Figure 2 overlays the phase diagrams. The dashed dividing line corresponds to Q's bid b_Q^1 and the solid line to b_Q^2. From the phase diagram, we see that $b_P^1 \prec b_Q^1 \prec b_P^2 \prec b_Q^2 \prec b_P^1$, and perturbing any of these bids a small amount would not change the outcome. If the mechanism were representable by a min function mechanism, then we would have $f_P(b_P^1) < f_Q(b_Q^1) < f_P(b_P^2) < f_Q(b_Q^2) < f_P(b_P^1)$, which is a contradiction.

This example is easily extended to two node-disjoint $s-t$ paths, each with at least 2 edges. If the bids on P are $x_1, \ldots, x_k$ and the bids on Q are $y_1, \ldots, y_\ell$, then in the phase diagram we represent b_Q by the dividing line with intercepts $y_1 + \frac{y_2}{2} + \sum_{i \geq 3} y_i$ and $\frac{y_1}{2} + y_2 + \sum_{i \geq 3} y_i$, and we represent b_P by the point $(x_1 + \sum_{i \geq 3} x_i, x_2)$. Thus, the extra bids on Q just move the dividing line outward and the extra bids on P just move the point to the right.

9 Truthfulness without independence, edge autonomy

The last section showed that some assumption on the structure of the graph is necessary to prove our characterization result. Here we show that the requirements of independence and edge autonomy are not vacuous conditions – that is, there exist truthful mechanisms for which these properties do not hold. First, we exhibit a truthful mechanism that satisfies sensitivity and edge and path autonomy but violates independence. Then we show a simple truthful mechanism for which the edge autonomy property fails. Both of these mechanisms are defined on a graph of three parallel arcs, so the graph falls into the classes covered by both Theorems 6.4 and 6.5.

Consider a graph consisting just of three parallel $s-t$ arcs, numbered 1, 2, and 3. If $b_1 + b_2 \leq 6b_3$, then the lower of b_1 and b_2 wins. Otherwise the lowest bid wins. Ties go to the lowest numbered bidder. This is clearly truthful and satisfies edge and path autonomy. Edges 1 and 2 are tied for the lead if $b_1 = b_2 \leq 3b_3$, because in this case edge 1 wins but any decrease to b_2 causes edge 2 to win. Since any increase in b_1 causes either b_2 or b_3 to win, the sensitivity property holds in this case. Edges 1 and 3 are tied for the lead if either ($b_1 + b_2 > 6b_3$ and $b_3 = b_1 \leq b_2$) or ($b_1 + b_2 = 6b_3$ and $b_3 \leq b_1 \leq b_2$), because in both cases b_1 wins, but any decrease in b_3 causes edge 3 to win. Sensitivity holds here, because any increase in b_1 causes either edge 2 or 3 to win. These are the only ways for edge 1 to be tied for the lead. A similar analysis holds for ties between edges 2 and 3. So the sensitivity property holds for this mechanism. If $b_1 = 2$, $b_2 = 3$ and $b_3 = 1$, then edge 1 wins, but if b_2 increases to 5, then edge 3 wins. Thus, the independence property fails.

It is also easy to construct a truthful mechanism for which edge autonomy fails. Consider the following path selection rule on the same graph as above. If $b_1 \leq b_2 + b_3$, then 1 wins. Otherwise, if $b_2 \leq b_3$ then 2 wins. Otherwise 3 wins. This mechanism is clearly monotone, hence truthful. When b_3 is fixed, there is no safe value for b_1 that is guaranteed to keep 1 from winning, because 2 can always cause 1 to win by bidding at least $b_1 - b_3$. Thus, edge autonomy fails for edge 1. The independence property also fails for this mechanism, as edge 2 can sometimes change the winner from 3 to 1 by raising her bid.

10 Efficient computation of min function mechanisms

To this point, we have not been concerned with how to compute min function mechanisms efficiently, since our goal was to prove negative results. However, even though all min function mechanisms can be forced to overpay just as badly as VCG in the worst case, we have reason to believe that certain classes of them might perform better in practice. Here, we mention one attractive and natural class that can be computed in polynomial time.

Our original motivation for defining min function mechanisms was to provide a means for biasing the path selection rule against paths with a large number of hops. The hope is that such a path selection rule will choose a path with fewer hops, so we will pay bonuses to fewer edges. Here is one way to introduce such a bias. Let g be any positive, increasing (and easily computed) function defined on the positive integers, and define $f_P(b_P) = g(|P|)\sum_{e \in P} b_e$, where $|P|$ denotes the number of edges in path P.

This class of min function mechanisms can be computed in polynomial time, using $O(n)$ Bellman-Ford shortest path computations. A single run of Bellman-Ford computes the shortest path from s to t using k hops, for each $k = 1, \ldots, n-1$. To compute the winning path, we just multiply the length of the shortest k-hop path by $g(k)$, and take the minimum result. To compute the threshold bid for each winning edge e, we perform another Bellman-Ford shortest path computation on the graph with edge e deleted, to determine which path Q has the minimum function value among all paths not using edge e. There may be several values k for which the shortest k-hop path P_k uses edge e, and for which $f_{P_k}(b_{P_k}) \le b_Q(b_Q)$. For each of these, Q becomes preferred once e has raised its bid by $(f_Q(b_Q) - f_{P_k}(b_{P_k}))/g(k)$. Computing the maximum of these increments gives the bonus for edge e, because that is the lowest bid for which a path not containing e wins.

Hershberger and Suri [12] demonstrate how to compute the winning path and all payments for the VCG mechanism in time dominated by two Dijkstra shortest path computations. It would be interesting if one could achieve a similar result for min function mechanisms of the form described above.

11 Characterization result for general team selection mechanisms

Min function mechanisms can be defined for the general problem of team selection in the obvious way: we just define a function f_P for every feasible team P. In the definitions of edge and path autonomy, independence and sensitivity, if we replace "edge" everywhere by "agent," and "path" by "team," we obtain the analogous properties for team selection mechanisms. In the proofs of Lemma 6.3 and Theorems 6.4 and 6.5, the only way in which we relied on the fact that the teams were paths is that no $s-t$ path is a subset of another. Our definition of the team selection problem already specified that no feasible team should be a subset of another. Thus, analogs of these three results all hold for the general team selection problem. The analog of a graph containing the edge $s-t$ is a set of feasible teams containing at least one team consisting of a single agent. The analog of a connected graph plus two extra $s-t$ paths disjoint from the rest of the graph is a set of feasible teams containing at least three teams, where two of the teams consist of agents belonging only to that team. Using these analogous concepts, we can prove analogs of Lemma 6.3 and Theorems 6.4 and 6.5. The proofs go through without modification. Theorem 7.1 did rely slightly on the structure of paths. The condition we need for an analog of this theorem to hold is that there are three feasible teams P, Q, and R such that the only feasible teams that are subsets of $P \cup Q \cup R$ are P, Q, and R.

12 Conclusion

Feigenbaum et al. [8] did some experiments to see how much the VCG mechanism overpays in some synthetic examples meant to model the problem of routing between autonomous systems on the Internet. They ran the mechanism on a 5773-node graph where the nodes denote autonomous systems from a recent snapshot of the Internet, and the edges denote links between them. In their model, the nodes incur costs rather than the links. They assumed all node costs were 1, and computed VCG prices for randomly selected $s - t$ pairs. They report that 68% of the node payments were 1, 28% of them were 2, and the average node payment was 1.44. This means that VCG overpaid by 44% on average. However, we believe that this number is artificially small because of the design of the experiment. Since the node costs were all taken to be 1, computing the shortest path is equivalent to computing the path with the least number of hops. If deleting a winning node results in an alternate path using the same number of hops, then this node's threshold bid is 1, so she receives no bonus. As they report, this occurred 68% of the time. If we were to add some variation in the node costs, all of these zero bonuses would now become positive. It would be interesting to test how significantly this changes their experimental results. In this experiment, using min function mechanisms that are biased towards paths with few hops would not help, because VCG is already choosing the path with the fewest hops. In an experiment where the node costs vary widely, the shortest paths could often have more than the minimum number of hops. In these cases, it would be interesting to test whether min function mechanisms could give improved results.

13 Acknowledgments

This paper appeared previously as an extended abstract [3].

References

[1] A. Archer, C. Papadimitriou, K. Talwar, and É. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. *Internet Mathematics*. To appear.

[2] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the* 42^{nd} *IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2001.

[3] A. Archer and É. Tardos. Frugal path mechanisms. In *Proceedings of the* 13^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 991–999, 2002.

[4] S. Bikhchandani, S. de Vries, J. Schummer, and R. V. Vohra. Linear programming and Vickrey auctions. In B. Dietrich and R. V. Vohra, editors, *Mathematics of the Internet: E-Auction and Markets*, IMA Volumes in Mathematics and its Applications, Volume 127, pages 75–116. Springer, New York, 2002.

[5] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[6] S. DeVries and R. Vohra. Combinatorial auctions: A survey. *INFORMS Journal of Computing*, 15:284–309, 2003.

[7] E. Elkind, A. Sahai, and K. Steiglitz. Frugality in path auctions. In *Proceedings of the* 15^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[8] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 2002 ACM Symposium on Principles of Distributed Computing*, pages 173–182, 2002.

[9] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, 1991.

[10] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[11] T. Groves. On theories of incentive compatible choice with compensation. In Hildenbrand [13], pages 1–29.

[12] J. Hershberger and S. Suri. Vickrey prices and shortest paths: What is an edge worth? In *Proceedings of the* 42^{nd} *IEEE Symposium on Foundations of Computer Science*, pages 252–259, 2001.

[13] W. Hildenbrand, editor. *Advances in Economic Theory*. Cambridge University Press, New York, 1982.

[14] V. Krishna. *Auction Theory*. Academic Press, New York, 2002.

[15] J.-J. Laffont and E. Maskin. The theory of incentives: An overview. In Hildenbrand [13], pages 31–94.

[16] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proceedings of the* 44^{th} *Annual IEEE Symposium on Foundations of Computer Science*, pages 574–583, 2003.

[17] A. Mas-Collel, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[18] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of the* 31^{st} *Annual ACM Symposium on Theory of Computing*, pages 129–140, 1999.

[19] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the* 2^{nd} *ACM Conference on Electronic Commerce*, pages 242–252, 2000.

[20] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[21] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.

[22] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the* 33^{rd} *Annual ACM Symposium on Theory of Computing*, pages 749–753, 2001.

[23] K. Roberts. The characterization of implementable choice rules. In J.-J. Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–348. North-Holland, Amsterdam, 1979.

[24] K. Talwar. The price of truth: Frugality in truthful mechanisms. In 20^{th} *Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, Volume 2607, pages 608–619. Springer, Heidelberg, 2003.

[25] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.