Cheap Labor Can Be Expensive

Ning Chen Anna R. Karlin

Department of Computer Science & Engineering,
University of Washington, Seattle, USA
{ning,karlin}@cs.washington.edu

Abstract

We study markets in which consumers are trying to hire a team of agents to perform a complex task. Each agent in the market prices their labor, and based on these prices, consumers hire the cheapest available team capable of doing the job they need done. We define the *cheap labor cost* in such a market as the ratio of the best Nash equilibrium of the original market and the best possible Nash equilibrium of any of its submarkets, where "best" is defined with respect to consumers, i.e., we are looking at Nash equilibria in which the consumer pays the least. This definition is motivated by a "Braess-style" paradox: in certain kinds of marketplaces, competition, in the form of the availability of "cheap labor", can actually cause the prices paid by consumers to go up.

We present tight bounds on the cheap labor cost for a variety of markets including s-t path markets, matroid markets and perfect bipartite matching markets. The differences in cheap labor cost across markets demonstrate the complex relationship between the combinatorial structure of the marketplace and the advantages or more precisely, disadvantages to consumers due to competition.

1 Introduction

It is common wisdom that competition between vendors of goods and services benefits, or at least never hurts, consumers. Intuitively, when faced with competition, a service provider may be forced to lower prices in order to maintain or increase their market share. Indeed, this is one of the key motivations for anti-monopoly laws.

Consider, for example, a customer wishing to purchase the rights to have data routed on its behalf from a source s to a sink t in a network where each link is owned by a different non-cooperative selfish agent. Given the pricing of the links in the network, the customer will naturally choose to purchase the cheapest route.

How do we expect agents (edges) to price their links? The price an agent chooses depends (primarily) on two things: their costs and the competition they face. Each agent has an associated overhead/cost it incurs for routing data, and thus agents will certainly set their prices to at least recover these costs. Moreover, since agents are selfish, they will try to set their prices much higher, in an attempt to maximize their own profit. Clearly, if there is only one path from s to t in the network, the edges on that path have a monopoly and can ask arbitrarily high prices (assuming the routing is necessary for the consumer). On the other hand, if there is competition in the form of another parallel path from s to t, edges on each path are limited in how high they can set their prices, because the edges on the alternative path can undercut them if they go too high.

The setting we have just described defines a game, in fact a *first-price auction*, in which the agents' strategies consist of the possible prices or *bids* they offer customers. Following standard practice, we consider a price vector in *Nash equilibrium* to be a rational outcome of the process. A set of agent bids is a Nash equilibrium if no agent has an incentive to unilaterally change its bid to get more profit[1], given the bids by the other agents. Since there may be many Nash equilibria and our goal is to understand the impact of competition on consumer surplus, we consider the best possible Nash equilibrium from the perspective of the consumer, namely the Nash equilibrium in which the consumer pays the least [1, 23]. We call such a Nash equilibrium the *best Nash equilibrium.*

Consider, for example, Figure 1 which shows a network with two paths from s to t. The values shown on the edges are the overheads/costs that each of the agents incurs to route data. It is easy to see that in this network, assuming ties are broken in favor of the bottom path, one of the possible Nash equilibria is when each of the two edges on the bottom path bids 1/2, and the upper edge bids 1.[2]

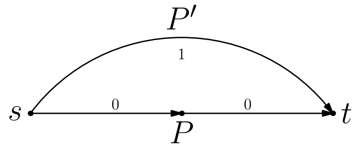


Figure 1: Two sets of edges P and P' compete to sell the s-t path.

Paradoxically, it is also possible for more competition to lead to a *reduction in consumer surplus* in

[1]The profit of an agent is the difference between the price he is paid by the consumer and his overhead, if he is selected, and 0 otherwise.

[2]If any of the edges on the bottom path raises its bid to try to make more money, they will "lose" the customer's business. On the other hand, the upper edge has no incentive to lower its bid, since it will run a deficit if it charges less than its overhead. Raising its bid doesn't benefit it either.

the path auction setting.[3] Consider Figure 2 where, again, the value on an edge is the overhead incurred by that edge if it routes data. In this example, the total consumer payment in the best Nash equilibrium is 10, where for example (e_1, e_3, e_6) is the winning path and e_1 and e_6 each bid 5, and all other edges bid their cost. However, if we eliminate the competition present due to the existence of edges e_3 and e_4, the best Nash equilibrium for the consumer results in a total payment of 5, where for example (e_1, e_5) is the winning path and all edges bid their cost.

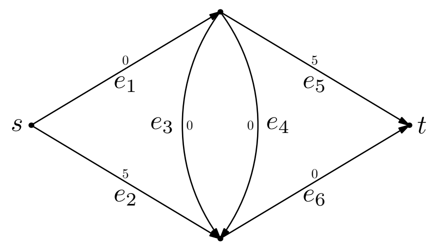


Figure 2: An example that competition leads to worse consumer surplus.

It is this intriguing example that motivates us to consider the following questions. How much consumer surplus can be lost due to competition by some "cheap" agents? Can we efficiently determine which agents' removal from the marketplace is most helpful to consumers? How do the answers to these questions depend on the combinatorial structure of the marketplace?

We study this problem by considering a quantity we call the *cheap labor cost* of a market, which is equal to the consumer payment in the best Nash equilibrium of the original market divided by the consumer payment in the best possible Nash equilibrium of any of its submarkets, where a submarket is just the market defined by some subset of agents of the original market. We call the submarket with the best possible Nash equilibrium the *consumer-optimal* submarket. In the example of Figure 2, the cheap labor cost is 2, and the consumer-optimal submarket is the submarket obtained by removing edges e_3 and e_4.

1.1 Our results

To study the cheap labor cost of markets as a function of their combinatorial structure, we adopt a general framework initiated in [3, 41, 23] of *hiring a team of agents*. Specifically, we consider a consumer wishing to hire a team of agents capable of performing a certain complex task on his behalf. Each agent e can perform a simple task/service at a cost (to himself) of $c(e)$. Based on their knowledge of the marketplace, each of the agents sets a price for his service – we refer to the price set by agent e as his bid $b(e)$ – and based on these bids, the consumer selects a *feasible set* — a set of agents whose combined services are sufficient to perform the complex task — and pays each selected agent their bid value. Naturally, the consumer wishes to pay as little as possible to get the job done and thus we assume that the consumer selects the feasible set whose combined services will cost him the least money. Thus, the particular setting is defined entirely by the *set system* of feasible sets and the costs the agents incur.

Two special cases of this general setting have been studied extensively in the past: (i) The *path auction* [30, 3, 15, 23], discussed above, where the agents own edges of a known graph, and the consumer wants to purchase a path between two given nodes s and t (or have data routed on its behalf.) (ii) The

[3] This is reminiscent of Braess's paradox [5, 29, 36, 26], where adding more edges to a network might hurt all of the traffic.

minimum spanning tree auction [41, 4], where the agents again own edges of a graph, and the consumer wants to purchase a minimum spanning tree.

Prior work in these settings has focused largely on the frugality of truthful mechanisms such as the VCG mechanism and on the design of truthful mechanisms with optimal frugality. Here, our goal is very different – we seek to understand the relationship between the combinatorial structure of the marketplace and the ramifications of competition from the perspective of the consumer.

Our main results are the following:

- In Section 3, we prove that the cheap labor cost of any market is upper bounded by the size of the cheapest feasible set, say n, and we give an algorithm for computing a Nash equilibrium that is within a factor of n of the best Nash equilibrium. While the bound we give on the cheap labor cost is not tight in general, we show that for certain interesting markets, such as the *perfect bipartite matching markets*, discussed in Section 4, it is.

- At the other end of the spectrum, in Section 5, we show that for spanning tree markets, and more generally for any set system where the feasible sets are bases of a matroid, the cheap labor cost is one. Thus, *matroid markets* are the canonical marketplaces in which competition and the existence of cheap laborers never hurt the consumer. We also give an algorithm for computing the best Nash equilibrium in a matroid markets.

- Finally, in Section 6, we return to the *s-t path markets* discussed above, and show that the cheap labor cost is at most two. In other words, the example given in Figure 2 is the worst possible case. We also show that finding the best Nash equilibrium of the consumer-optimal submarket is NP-hard, and even approximating the price the consumer pays in this optimal submarket to within a factor better than two is NP-hard. In the process, we are also able to show that approximating the best Nash equilibrium in the original market to within a factor better than two is NP-hard. On the other hand, we show that it is easy to get a 2-approximation to the best Nash equilibrium.

1.2 Related Work

The effectiveness, efficiency and social ramifications of competitive markets are objects of extensive study in economics. Most research in this area has focused on the operation of markets, demand and supply analysis, social welfare, competitive equilibria, and so on. See, e.g., [27] for a comprehensive discussion. To the best of our knowledge, no previous work in the economics literature studies the cheap labor cost we study here measuring loss in consumer surplus due to competition in the market.

The problem of hiring a team of agents in a complex setting at minimum total cost to the consumer, has been shown to have many practical economic applications and has been studied extensively [30, 3, 15, 41, 22, 14, 23, 11, 34]. The path auction problem in particular has been received a great deal of attention. Most of the work in this area has been on the study of truthful auction mechanisms, such as the VCG mechanism [44, 7, 20], and on understanding the frugality of truthful mechanisms. In addition, path auctions have been studied from the Bayesian perspective [15, 11]. There are also some papers which consider Nash equilibria in first-price path auctions. For example, motivated by the large overpayments required for truthful mechanisms, Immorlica, Karger, Nikolova and Sami [22] study Nash equilibria in first-price auctions, with a particular focus on the overpayments compared to VCG. Karlin, Kempe and Tamir [23] propose the payments in a best Nash equilibrium for a first-price auction as a benchmark against which to evaluate the payments of truthful mechanisms.

There have been papers which considered the question of how to best remove edges from a network to improve efficiency. For example, Roughgarden [36] studied the complexity of finding the best subnetwork to be used for selfish routing when the goal is to minimize total latency. Elkind [14] showed that VCG payments can sometimes be improved by a factor of $\Theta(n)$ for the path auction problem if one can remove edges from the network.

Finally, one can also view this research as vaguely similar in spirit to work on the price of anarchy [24, 39, 38, 28, 35, 10, 43, 12, 17, 21] and the price of stability [1, 2, 8] which quantifies loss of efficiency due to selfish behavior. See Papadimitriou [33] and Tardos [42] for recent surveys on these topics.

2 Preliminaries

We consider a consumer wishing to hire a team of agents capable of performing a certain complex task on his behalf. Following [41, 23], the consumer has access to a market $\mathcal{M}$ defined by a set system $(E, F; c)$, where $E = \{e_1, \ldots, e_n\}$ is a collection of agents, F is a collection of *feasible* subsets of E and c is a cost vector for agents. A subset $S \subseteq E$ is *feasible* (i.e., $S \in F$) if the agents in S have the combined skills necessary to accomplish the consumer's task. We assume there are m *feasible* combinations of agents, $F = \{S_1, \ldots, S_m\}$. We also assume that (a) each agent $e \in E$ has an associated *cost* $c(e) \in \mathbb{N}$, representing the cost that agent incurs (in time, money or resources, measured in terms of some standard currency) to provide his labor, and (b) each agent's goal is to make as large a profit as possible. An agent's profit is the difference between what he is paid and the cost he incurs to provide his services.

A basic question is: what will the consumer have to pay to get the job done? We assume that each agent $e \in E$ submits a *bid*, $b(e) \in \mathbb{R}$, representing the price at which the agent is willing to provide his services. Based on the bids of agents, the consumer will hire the cheapest feasible subset of agents (i.e., the feasible subset $S \in F$ such that $\sum_{e \in S} b(e)$ is minimized) to do the job. The consumer will pay each of the agents hired their bid value. Thus, the consumer is in essence running a *first-price auction*. We will refer to the set of agents hired as the *winning set* and each agent in this set as a *winning agent*. If $e \in E$ is a winning agent, its *profit* is $b(e) - c(e)$; otherwise, its profit is zero. In this paper we assume $b(e) \geq c(e)$, for $e \in E$, since no agent aspires to obtain a negative profit.

The model just described defines a game, in which an agent's strategy set consists of the possible bids it can propose to the consumer. We say a bid vector b is a *Nash equilibrium* (NE) if no agent has an incentive to unilaterally change its bid to get more profit, given the bids by the other agents. As we shall see shortly, this game has pure Nash equilibria, and since in this context, it is natural to imagine that bidding is performed through posted prices and agents can see each other's bids, mixed strategies are not terribly natural. Thus, we restrict attention in this paper only to pure equilibria. We define the *value* of a NE given by a bid vector b to be $b(S) = \sum_{e \in S} b(e)$, where $S \in F$ is the winning set w.r.t b. A NE b is said to be a *best Nash equilibrium* if, among pure Nash equilibria, it minimizes the total price paid by the consumer. Thus, the best NE represents an equilibrium price paid by the consumer which *maximizes the consumer's surplus.*

2.1 A Greedy Algorithm to Compute a Nash Equilibrium

If there is an agent $e \in E$ that holds a monopoly, i.e., $e \in S$ for every feasible set $S \in F$, and the consumer must buy a feasible set, no Nash equilibrium exists, since e can increase its bid arbitrarily. Thus, we assume that the market is *monopoly-free*. In this case, the existence of a pure NE follows trivially by running the following greedy algorithm.[4]

```
Algorithm 1.
1.  Find the cheapest feasible set S ∈ F w.r.t costs
    (breaking ties, say, lexicographically).
2.  For each e ∈ E, initialize b(e) to c(e).
3.  For each e ∈ S
4.     raise b(e) until there is S' ∈ F s.t.  e ∉ S' and b(S) = b(S').
5.  Output the bid vector b and winning set S.
```

It is easy to verify the following proposition:

Proposition 2.1 *The bid vector b and winning set S generated by* `Algorithm 1` *define a pure NE.*[5]

Proof. For any agent $e \in E \setminus S$, since $b(e) = c(e)$, e has no incentive to change its bid to obtain more profit. For any agent $e \in S$, consider the threshold that we cannot increase $b(e)$ any more in Step 4 of the algorithm. At that time, there is $S' \in F$ such that $e \notin S'$ and $b(S) = b(S')$. That is, $b(S \setminus S') = b(S' \setminus S) = c(S' \setminus S)$. Note that we cannot increase the bids of agents in $S \setminus S'$ after that point, which implies that at the end of the algorithm, we still have $b(S \setminus S') = b(S' \setminus S)$. Thus, agent e does not have an incentive to change its bid. □

We will need the following general fact about Nash equilibria.

Proposition 2.2 *Consider any NE b where $S \in F$ is the winning set. For any $e \in S$, there is another feasible set $S' \in F$, $e \notin S'$, such that $b(S \setminus S') = b(S' \setminus S) = c(S' \setminus S)$. We call such a set S' a* tight set *w.r.t b.*

Proof. First, due to the first-price auction, we know $b(S \setminus S') \le b(S' \setminus S)$. If $b(S \setminus S') < b(S' \setminus S)$ for all feasible set S' where $e \notin S'$, agent e can raise its bid by a small amount and still win the auction, which contradicts the assumption that b is a NE.

Now consider S' such that $e \in S$, $e \notin S'$ and $b(S \setminus S') = b(S' \setminus S)$. If $b(S' \setminus S) > c(S' \setminus S)$, then some agent in $S' \setminus S$ can reduce its bid a little to win the auction, again contradicting the assumption that b is a NE. □

2.2 Cheap Labor Cost

Given a market $\mathcal{M} = (E, F; c)$, we say $\mathcal{M}' = (E', F'; c)$ is a *submarket* of $\mathcal{M}$, denoted by $\mathcal{M}' \subseteq \mathcal{M}$, if $E' \subseteq E$ and $F' = \{S \in F \mid S \subseteq E'\}$. That is, $\mathcal{M}'$ is generated from $\mathcal{M}$ by removing some agents and keeping the remaining agents with their associated costs and the corresponding feasible sets.

Definition 2.1 ***(Cheap Labor Cost)*** *For any market $\mathcal{M}$, let $\nu_{\mathcal{M}}$ be the value of the best NE of $\mathcal{M}$. The* cheap labor cost *of $\mathcal{M}$ is defined by*

$$\frac{\nu_{\mathcal{M}}}{\min_{\mathcal{M}' \subseteq \mathcal{M}} \nu_{\mathcal{M}'}}$$

[4]Note that when we talk about computing NE, we are assuming full information about the agents' costs. We are explicitly not studying the question of how the agents would arrive at a Nash equilibrium, which is of course a very interesting question in its own right. We are trying merely to understand the impact of various Nash equilibria on consumer surplus.

[5]Technically, this requires that the consumer breaks ties among the sets with the cheapest bid value the same way ties are broken in Step 1 of `GreedyAlg`. Alternatively, one can avoid tie-breaking issues completely by using the notion of an ϵ-Nash equilibrium. See [22] for a discussion of some of the subtleties involved.

where $\nu_{\mathcal{M}'}$ is the value of the best NE of submarket $\mathcal{M}'$. If there is no feasible solution in $\mathcal{M}'$, we define $\nu_{\mathcal{M}'}$ as infinity. We use $\mathcal{M}^* \triangleq \arg\min_{\mathcal{M}' \subseteq \mathcal{M}} \nu_{\mathcal{M}'}$ to denote the consumer-optimal *submarket.*

Note that since the whole market is itself a submarket, the cheap labor cost of any market is at least one. The cheap labor cost characterizes how much better off the market can be for the consumer by removing a subset of the agents.

In the following sections we will study the cheap labor cost for different types of markets.

3 A General Upper Bound on the Cheap Labor Cost

We begin by proving a general upper bound on the cheap labor cost of any market.

Theorem 3.1 *The cheap labor cost of any market $\mathcal{M} = (E, F; c)$ is at most $|S|$, where $S \in F$ is a feasible set with the minimum total cost.*

Proof. Let b be a NE of $\mathcal{M}$ computed by `Algorithm 1`, and S the corresponding winning set. Consider any submarket $\mathcal{M}' \subseteq \mathcal{M}$. Assume b' is a best NE of $\mathcal{M}'$ and $S' \in F'$ is the winning set. It suffices to show that $b(S) \leq |S| \cdot b'(S')$.

Note that for any agent $e \in S' \setminus S$, e is not a winner in the NE computed by `Algorithm 1`. Thus, we have $b(e) = c(e)$ and $b(S' \setminus S) = c(S' \setminus S)$, which implies that $c(S' \setminus S)$ is an upper bound that agents in $S \setminus S'$ can bid up to. That is,

$$b(S \setminus S') \leq b(S' \setminus S) = c(S' \setminus S) \leq b'(S' \setminus S)$$

For each $e \in S \cap S'$, according to Proposition 2.2, in the best NE b' of $\mathcal{M}'$, there is a set $S'' \in F'$ such that $b'(S') = b'(S'')$ and $e \notin S''$. We claim that for each such e, $b(e) \leq b'(S')$. Otherwise, we have $b(S) = b(S \setminus S'') + b(S \cap S'') > b'(S') + b(S \cap S'') = b'(S'') + b(S \cap S'') \geq c(S'') + b(S \cap S'') \geq c(S'' \setminus S) + b(S \cap S'') = b(S'' \setminus S) + b(S \cap S'') = b(S'')$ which contradicts the fact that S is the winning set according to bid vector b.

Therefore, $b(S) = b(S \setminus S') + b(S \cap S') \leq b'(S' \setminus S) + |S \cap S'| \cdot b'(S') \leq |S| \cdot b'(S')$, completing the proof of the theorem. □

The following corollary is immediate.

Corollary 3.1 *For any market $\mathcal{M} = (E, F; c)$,* `Algorithm 1` *computes a NE whose value is an $|S|$-approximation to the best NE of both $\mathcal{M}$ and $\mathcal{M}^*$, where $S \in F$ is a feasible set with the minimum total cost.*

In general, this theorem is far from tight. However, there are certain market types for which it is tight. An example is given in the next section.

4 The Perfect Bipartite Matching Market

In a *perfect bipartite matching market*, the agents E represent the set of edges in a bipartite graph $G = (U, V; E)$, where U and V are two disjoint sets of vertices, $|U| = |V|$, and all edges have one endpoint in U and one endpoint in V. A subset $S \subseteq E$ is feasible if the edges in S form a perfect matching in G. Since the size of each feasible set is $|V|$, by Theorem 3.1, the cheap labor cost of the perfect bipartite matching market is at most $|V|$.

We now show that in the worst case, this bound is essentially tight. Consider the market shown in Figure 3. This market is given by $G = (U, V; E)$, where $U = \{u_1, u_2, u_{2'}, \ldots, u_k, u_{k'}\}$ and $V = \{v_1, v_2, v_{2'}, \ldots, v_k, v_{k'}\}$, and

$$E = \{(u_1, v_1)\} \cup \{(u_1, v_i), (u_i, v_1), (u_{i'}, v_1), (u_i, v_{i'}), (u_i, v_i), (u_{i'}, v_{i'}) \mid i = 2, \ldots, k\}.$$

The cost of edge $(u_{i'}, v_1)$ is one, $i = 2, \ldots, k$, and the cost of any other edge is zero. It is not hard to check that the consumer-optimal subgraph is $G' = (U, V; E')$, where $E' = E \setminus \{(u_i, v_1) \mid i = 2, \ldots, k\}$. Let $H = \{(u_1, v_1)\} \cup \{(u_i, v_i), (u_{i'}, v_{i'}) \mid i = 2, \ldots, k\}$. In G', the best NE has a value of one, where edge (u_1, v_1) bids one and all other edges in $E' \setminus \{(u_1, v_1)\}$ bid their cost, and H is the winning perfect matching. On the other hand, the value of the best NE of G is $k - 1$, where each edge $(u_{i'}, v_{i'})$ bids one, $i = 2, \ldots, k$, and all other edges bid their cost, and H is the winning perfect matching. Therefore, the cheap labor cost is $k - 1 = O(|V|)$.

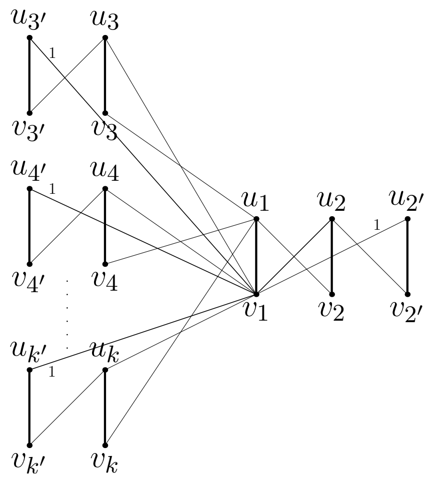


Figure 3: A tight example of cheap labor cost of perfect bipartite matching.

5 Matroid Markets

In this section we consider *matroid markets* $\mathcal{M} = (E, F; c)$, where E is the set of elements of a matroid and F is the set of bases of the matroid. A natural special case is when the set E represents the edges in a graph $G = (V, E)$ and the feasible sets are spanning trees in the graph. In this case, Theorem 3.1 guarantees that the cheap labor cost of the market is at most $|V| - 1$, where $|V|$ is the number of vertices in the graph. In contrast to the perfect bipartite matching market, this upper bound is way off. In fact, we will show in this section that for spanning tree markets and any market where the feasible sets are bases of a matroid, the cheap labor cost is one.

We use the following well-known property of matroids [6].

Proposition 5.1 *A collection $F = \{S_1, \ldots, S_m\}$ of feasible sets is the set of bases of a matroid if and only if for any $S_i, S_j \in F$, there is a bijection f between $S_i \setminus S_j$ and $S_j \setminus S_i$ such that $S_i \setminus \{e\} \cup \{f(e)\} \in F$, for any $e \in S_i \setminus S_j$.*

We begin by characterizing the winning set for matroid markets.

Lemma 5.1 *For any submarket $\mathcal{M}' = (E', F'; c)$ of a matroid market $\mathcal{M} = (E, F; c)$, the winning set in any NE of $\mathcal{M}'$ is one with the minimum total cost among all feasible sets in F'.*

Proof. Let S_i be the winning set of a NE b' of $\mathcal{M}'$. Thus, $b'(S_i) \le b'(S_j)$, for any $S_j \in F'$. Let f be the bijection between $S_i \setminus S_j$ and $S_j \setminus S_i$ given by the above lemma. Note that $S_i \cup S_j \subseteq E'$. For any $e \in S_i \setminus S_j$, since $S_i \setminus \{e\} \cup \{f(e)\} \in F$ and $S_i \setminus \{e\} \cup \{f(e)\} \subseteq E'$, we know $S_i \setminus \{e\} \cup \{f(e)\} \in F'$ is a feasible set of $\mathcal{M}'$. Thus, $b'(S_i) \le b'(S_i \setminus \{e\} \cup \{f(e)\})$. Hence, $c(e) \le b'(e) \le c(f(e))$, otherwise agent $f(e)$ would have an incentive to reduce its bid to be the winner. Therefore,

$$\begin{aligned} c(S_i) &= c(S_i \cap S_j) + \sum_{e \in S_i \setminus S_j} c(e) \le c(S_i \cap S_j) + \sum_{e \in S_i \setminus S_j} c(f(e)) \\ &= c(S_i \cap S_j) + c(S_j \setminus S_i) = c(S_j). \end{aligned}$$

The lemma follows. □

Theorem 5.1 *The cheap labor cost is one for any matroid market $\mathcal{M}$.*

Proof. Let $E = \{e_1, \ldots, e_n\}$ and feasible set $F = \{S_1, \ldots, S_m\}$. Assume $c(S_1) \le c(S_2) \le \cdots \le c(S_m)$. Let b be a NE of $\mathcal{M}$ computed by `Algorithm 1` with S_1 being the winning set.

Consider any submarket $\mathcal{M}' = (E', F'; c)$ of $\mathcal{M}$, where $E' \subseteq E$ and $F' \subseteq F$. Assume b' is a best NE of $\mathcal{M}'$ and $S' \in F'$ is the winning set. Note that according to Lemma 5.1, for any $S'' \in F'$, $c(S') \le c(S'')$.

Next we will compare $b(S_1)$ with $b'(S')$. Note that since $c(S_1) \le c(S')$, according to `Algorithm 1`, we have

$$b(S_1 \setminus S') \le b(S' \setminus S_1) = c(S' \setminus S_1) \le b'(S' \setminus S_1)$$

For any $e \in S_1 \cap S'$, consider the best NE b' of $\mathcal{M}'$. Since S' is the winning set of b' and $e \in S'$, we know there is $S'' \in F'$ such that (i) $e \notin S''$, (ii) $b'(S') = b'(S'')$, and (iii) $S' \setminus S'' = \{e\}$. Note that the first two conditions follow from Proposition 2.2, and the last condition follows from Proposition 5.1. Let $S'' \setminus S' = \{e'\}$. We know $b'(e) = b'(e') = c(e')$. Consider the two feasible sets S_1 and S'' in $\mathcal{M}$. By the symmetric basis-exchange axiom [6, 32], we know there is $e'' \in S''$ such that $S_1 \setminus \{e\} \cup \{e''\} \in F$ and $S'' \setminus \{e''\} \cup \{e\} \in F$. If $e'' = e'$ (which implies that $e' \notin S_1$), according to `Algorithm 1`,

$$b(e) \le c(e') = b'(e)$$

If $e'' \neq e'$, we know $e'' \in S' \cap S''$. Since all elements of $S'' \setminus \{e''\} \cup \{e\}$ are in E', $S'' \setminus \{e''\} \cup \{e\}$ is a feasible set of F'. Thus, $c(S') \le c(S'' \setminus \{e''\} \cup \{e\})$, which implies that $c(e'') \le c(e')$. On the other hand, we know $S_1 \setminus \{e\} \cup \{e''\} \in F$. Again due to `Algorithm 1`,

$$b(e) \le c(e'') \le c(e') = b'(e)$$

Therefore,

$$b(S_1) = b(S_1 \setminus S') + \sum_{e \in S_1 \cap S'} b(e) \le b'(S' \setminus S_1) + \sum_{e \in S_1 \cap S'} b'(e) = b'(S')$$

Since the value of the best NE of $\mathcal{M}$ is at most $b(S_1)$, which is smaller than or equal to the value of the best NE of any of its submarkets. Thus, the cheap labor cost of $\mathcal{M}$ is one. □

Note that in the proof of the above theorem, we essentially showed that the NE computed by `Algorithm 1` has the minimum value. Thus, we have the following conclusion.

Corollary 5.1 *For any matroid market $\mathcal{M} = (E, F; c)$,* `Algorithm 1` *computes a best NE of $\mathcal{M}$.*

An alternative proof of Theorem 5.1 can be obtained by using the facts that (1) VCG has frugality ratio one for matroid markets [23] and (2) VCG payments cannot increase with the addition of agents for matroid markets (or more precisely, when the "agents are substitutes" property holds) [45].

6 Path Markets

The final class of markets we explore are those that arise from first-price path auctions, henceforth called *path markets*. In this case, the market is a connected directed graph $G = (V, E)$, where each edge corresponds to an agent. There are two special vertices $s, t \in V$. The consumer's goal is to purchase a path (or have data routed on its behalf) from s to t.

We begin with an important characterization about NE in path markets.

Lemma 6.1 *Let b be a NE for a path market G, with winning path P. Then there is a pair of edge-disjoint s-t paths P' and P'' such that $b(P') = b(P'') = b(P)$.*

For example, in the NE for the path market of Figure 2 (with $b(e_1) = b(e_6) = 5$, and $b(e) = c(e)$ for all other edges, and winning path (e_1, e_3, e_6)), the two paths P' and P'' are (e_1, e_5) and (e_2, e_6), and $b(P') = b(P'') = b(P) = 10$.

6.1 Proof of Lemma 6.1

The following claim follows from Proposition 2.2, where the statement is in terms of path markets.

Claim 6.1 *For any $e \in P$, there is another s-t path P', $e \notin P'$, such that $b(P \backslash P') = b(P' \backslash P) = c(P' \backslash P)$. We call such a path P' a* tight path *w.r.t b.*

Claim 6.2 *Let v be a vertex on a tight path P'. Then the portion of the prefix of the path P' from s to v is a cheapest path w.r.t b.*

Proof. Since P' is tight, $b(P') = b(P)$. If there is a shorter path from s to v than the prefix of P' from s to v, then there is a shorter path from s to t than $b(P)$, contradicting the fact that P is the winning path. ■

Claim 6.3 *Let T be the union of all edges on tight paths w.r.t b together with the edges in P. Every s-t path in T has total bid value $b(P)$.*

Proof. Let $\tilde{P}$ be an s-t path in T and let v be the first vertex on $\tilde{P}$ such that the prefix of $\tilde{P}$ terminating at v has total bid value greater than the length of the cheapest path from s to v (w.r.t b). Suppose that the last edge on this subpath is (u, v). Then the prefix of $\tilde{P}$ to u is a cheapest path to u. Moreover the edge (u, v) is on some tight path P'. Therefore due to Claim 6.2, the prefix of P' to u (and v) is a cheapest path to u (and v). That is, $\sum_{e \in (s,u)_{\tilde{P}}} b(e) = \sum_{e \in (s,u)_{P'}} b(e)$, where $(s, u)_P$ denotes the path from s to u along P, which implies that

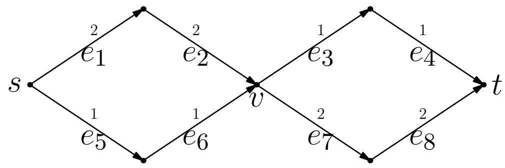
$$\begin{aligned} \sum_{e \in (s,v)_{\tilde{P}}} b(e) &= \sum_{e \in (s,u)_{\tilde{P}}} b(e) + b(u, v) \\ &= \sum_{e \in (s,u)_{P'}} b(e) + b(u, v) \\ &= \sum_{e \in (s,u)_{P'}} b(e) \end{aligned}$$

contradicting the assumption that the prefix of $\tilde{P}$ up to v is not a cheapest path. ■

We complete the proof of Lemma 6.1 by observing that by Claim 6.1, the set T is two-connected, and therefore, contains two edge-disjoint s-t paths. By Claim 6.3, these two paths both have bid value $b(P)$.

6.2 Cheap Labor Cost of Path Markets

We now turn to the main results about the cheap labor cost for path markets. We begin with a characterization of the best NE of the consumer-optimal subgraph. For any two edge-disjoint s-t paths P_1 and P_2, let $\delta(P_1, P_2)$ be the length of the longest s-t path w.r.t costs in the subgraph $P_1 \cup P_2$. In Figure 4, for example, $\delta(P_1, P_2) = 8$.



$P_1 = \{e_1, e_2, e_3, e_4\}$

$P_2 = \{e_5, e_6, e_7, e_8\}$

Figure 4: An example of definition $\delta(P_1, P_2)$.

Lemma 6.2 *Let P_1 and P_2 be two edge-disjoint s-t paths such that $\delta(P_1, P_2)$ is minimized. Then $\min_{G' \subseteq G} \nu_{G'} = \delta(P_1, P_2)$, where $\nu_{G'}$ is the value of the best NE of subgraph $G' \subseteq G$.*

Proof. Consider the subgraph $G'' = P_1 \cup P_2$. By abusing notations, let P_1 and P_2 be the shortest and longest s-t paths w.r.t costs in G'', respectively, and edge-disjoint. It is easy to verify that the best NE b'' of G'' with winning path P_1 satisfies $b''(P_1) = b''(P_2) = c(P_2) = \delta(P_1, P_2)$. Thus, $\min_{G' \subseteq G} \nu_{G'} \leq \delta(P_1, P_2)$.

Let $G^* = \arg\min_{G' \subseteq G} \nu_{G'}$ and b^* be a best NE of G^* with winning path P. By Lemma 6.1, G^* contains two edge-disjoint s-t paths P_1^* and P_2^* such that $b^*(P_1^*) = b^*(P_2^*) = b^*(P)$ (again by abusing notations, let P_1^* and P_2^* be the shortest and longest s-t paths w.r.t costs in $P_1^* \cup P_2^*$, respectively, and edge-disjoint). Observe that $b^*(P_1^*) \geq c(P_1^*)$ and $b^*(P_2^*) \geq c(P_2^*)$. Hence, $\min_{G' \subseteq G} \nu_{G'} = b^*(P) \geq \delta(P_1^*, P_2^*) \geq \delta(P_1, P_2)$.

Therefore, $\min_{G' \subseteq G} \nu_{G'} = \delta(P_1, P_2)$ and the lemma follows. ■

Theorem 6.1 *The cheap labor cost of any path auction problem is at most two.*

Proof. Let P_1 and P_2 be two edge-disjoint s-t paths such that $\delta(P_1, P_2)$ is minimized. Observe that $c(P_1) + c(P_2)$ gives an upper bound of ν_G. Thus, $\nu_G \leq c(P_1) + c(P_2) \leq 2\delta(P_1, P_2) = 2\min_{G' \subseteq G} \nu_{G'}$, yielding the desired result. ■

Essentially, the above theorem tells us that no matter how we delete edges from the original graph, the value of the best NE in the resulting subgraph is at least one half of that in the original graph. By the example from the introduction, we know that this bound is tight.

6.3 Complexity of the Cheap Labor Cost

A natural question to ask is whether, given a path market, we can find its best NE, and the best NE of its consumer-optimal submarket efficiently. In this section, we outline the arguments showing that both of these problems are NP-hard, and that in fact they are both hard to approximate to within a factor better than 2. The basis for the hardness of these results is the characterization of Lemma 6.2 and the following two polynomial time reductions.

Theorem 6.2 *For any directed graph G, finding two edge-disjoint s-t paths P_1 and P_2 such that $\delta(P_1, P_2)$ is minimized is NP-hard to approximate within a factor better than 2.*

Proof. We reduce from the following problem: Given a directed graph $G = (V, E)$, find two vertex-disjoint s-t paths P_1 and P_2 such that $\max\{c(P_1), c(P_2)\}$ is minimized. It was shown in [25] that this problem is NP-hard to approximate within a factor better than 2.

Given graph $G = (V, E)$, we construct a new graph G' as follows: for each $v \in V \setminus \{s, t\}$, we replace v in G' by an edge (v', v'') and connect (u'', v') if $(u, v) \in E$ (if $u = s$, connect (s, v'); and if $v = t$, connect (u'', t)). Let $c(v', v'') = 0$ for any $v \in V \setminus \{s, t\}$ and $c(u'', v') = c(u, v)$ for any $(u, v) \in E$. For example, the following graph is the construction of G' from Figure 4.

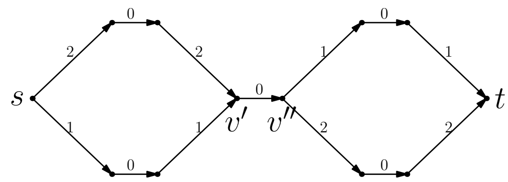


Figure 5: An example of construction graph G'.

For any s-t path $P = \{s = v_0, v_1, v_2 \ldots, v_k = t\}$ in G, define mapping

$$f(p) = \{s = v_0, v_1', v_1'', v_2', v_2'', \ldots, v_{k-1}', v_{k-1}'', v_k = t\}$$

It is easy to check $f(P)$ is an s-t path in G' and $c(P) = c(f(P))$.

For any vertex-disjoint s-t paths P_1 and P_2 in G, observe that $f(P_1)$ and $f(P_2)$ define two edge-disjoint (and vertex-disjoint as well) s-t paths in G'. Thus, $\delta(f(P_1), f(P_2)) = \max\{c(f(P_1)), c(f(P_2))\} = \max\{c(P_1), c(P_2)\}$. On the other hand, for any two edge-disjoint s-t paths P_1' and P_2' in G', according to the construction of G', note that essentially P_1' and P_2' are vertex-disjoint. Let P_1 and P_2 be the s-t paths in G such that $f(P_1) = P_1'$ and $f(P_2) = P_2'$. It is easy to see that such P_1 and P_2 exist and are vertex-disjoint. Thus, $\max\{c(P_1), c(P_2)\} = \max\{c(P_1'), c(P_2')\} = \delta(P_1', P_2')$.

Therefore, we know the two problems are equivalent and have the same optimal solution. Hence, our problem is NP-hard to approximate within a factor better than 2. ∎

Theorem 6.3 *There is a polynomial time reduction from the problem of computing the value of the best NE of the consumer-optimal subgraph to the problem of computing the value of the best NE of the original graph.*

Proof. Let $G = (V, E)$ be a path auction instance. Let $G^* = (V, E^*)$ be the consumer-optimal subgraph of $G = (V, E)$ where $E^* \subseteq E$, and b^* be a best NE of G^* with winning path P^*.

We construct a new graph G' according to G as follows: For each edge $(u, v) \in E$, we add a new vertex w and replace (u, v) by two new edges $(u, w), (w, v)$. Let the cost of (u, w) and (w, v) be 0 and $c(u, v)$, respectively, where $c(u, v)$ is the cost of (u, v) in G, as the following figure shows.

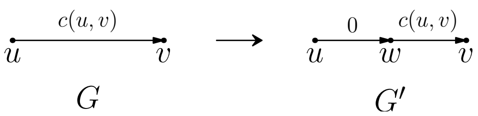


Figure 6: Construction of graph G'

Let b' be a best NE of G' with winning path P'. We claim that the value of the best NE of G' is equal to that of G^*, i.e., $b^*(P^*) = b'(P')$.

Given b^* of G^*, we construct a bid vector b_1 for G' as follows: For any $(u, v) \in E^*$, let $b_1(u, w) = 0$ and $b_1(w, v) = b^*(u, v)$; and for any $(u, v) \in E \setminus E^*$, let $b_1(u, w)$ and $b_1(w, v)$ be sufficiently large, where (u, w) and (w, v) in G' are the two corresponding edges of (u, v) in G. Let the winning path w.r.t b_1 in G' be P^*. It is easy to see that b_1 defines a NE for G'. Thus, $b'(P') \leq b_1(P^*) = b^*(P^*)$.

Given b' of G', let T be the union of all edges on tight paths w.r.t b' together with the edges in P'. We construct a bid vector b_2 for subgraph $G'' = (V, T)$ of G as follows: For each $(u, v) \in T$, let $b_2(u, v) = b'(u, w) + b'(w, v)$, where (u, w) and (w, v) in G' are the two corresponding edges of (u, v) in G''. Let the winning path w.r.t b_2 in G'' be P'. Note that b_2 defines a NE for G''. Thus, we have $b^*(P^*) \leq b_2(P') = b'(P')$. ■

Putting these facts together we obtain:

Theorem 6.4 *There are no polynomial time approximation algorithms for computing the best NE of path markets or for computing the best NE of the consumer-optimal subgraph with approximation ratio less than two unless P=NP.*

On the other hand:

Theorem 6.5 *For any path market on graph G,* `GreedyAlg` *generates a NE whose value is a 2-approximation to the best NE of both the original graph and consumer-optimal subgraph.*

Proof. For graph $G = (V, E)$, we run `GreedyAlg` on path P and get a NE b of G, where P is the cheapest s-t path w.r.t costs. Note that $b(e) = c(e)$ for any $e \in E \setminus P$. Let P_1 and P_2 be two edge-disjoint s-t paths minimizing $\delta(P_1, P_2)$. Thus, $b(P \setminus P_1) \leq b(P_1 \setminus P) = c(P_1 \setminus P)$ and $b(P \setminus P_2) \leq b(P_2 \setminus P) = c(P_2 \setminus P)$. Since $P \subseteq (P \setminus P_1) \cup (P \setminus P_2)$, we have $b(P) \leq c(P_1) + c(P_2) \leq 2\delta(P_1, P_2) = 2 \min_{G' \subseteq G} \nu_{G'} \leq 2\nu_G$. ■

Acknowledgements

We thank Rakesh Vohra for pointing out the alternative proof of Theorem 5.1. Thanks to Qiqi Yan for pointing out a mistake in an earlier draft of the paper.

References

[1] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, T. Roughgarden, *The Price of Stability for Network Design with Fair Cost Allocation*, FOCS 2004, 295-304.

[2] E. Anshelevich, A. Dasgupta, É. Tardos, T. Wexler, *Near-Optimal Network Design with Selfish Agents*, STOC 2003, 511-520.

[3] A. Archer, É. Tardos, *Frugal Path Mechanisms*, SODA 2002, 991-999.

[4] S. Bikhchandani, S. de Vries, J. Schummer, R. Vohra, *Linear Programming and Vickrey Auctions*, Mathematics of the Internet: E-auction and Markets (B. Dietrich and R. Vohra, editors), V127, 75-116, 2002.

[5] D. Braess, *Uber ein paradoxon der verkehrsplanung*, Unternehmensforschung, V.12, 258-268, 1968.

[6] R. A. Brualdi, *Comments on Bases in Dependence Structures*, Bulletin of the Australian Mathematical Society, V.1, 161-167, 1969.

[7] E. H. Clarke, *Multipart Pricing of Public Goods*, Public Choice, V.11, 17-33, 1971.

[8] J. R. Correa, A. S. Schulz, N. E. Stier Moses, *Selfish Routing in Capacitated Networks*, Mathematics of Operations Research, V.29(4), 961-976, 2004.

[9] P. Cramton, Y. Shoham, R. Steinberg, editors, *Combinatorial Auctions*, MIT Press, 2006

[10] A. Czumaj, P. Krysta, B. Vöcking, *Selfish Traffic Allocation for Server Farms*, STOC 2002, 287-296.

[11] A. Czumaj, A. Ronen, *On the Expected Payment of Mechanisms for Task Allocation*, PODC 2004.

[12] N. Devanur, N. Garg, R. Khandekar, V. Pandit, A. Saberi, V. Vazirani, *Price of Anarchy, Locality Gap, and a Network Service Provider Game*, WINE 2005, 1046-1055.

[13] P. Dubey, *Inefficiency of Nash Equilibria*, Mathematics of Operations Research, V.11, 1-8, 1986.

[14] E. Elkind, *True Costs of Cheap Labor are Hard to Measure: Edge Deletion and VCG Payments in Graphs*, EC 2005, 108-116.

[15] E. Elkind, A. Sahai, K. Steiglitz, *Frugality in Path Auctions*, SODA 2004, 701-709.

[16] D. Eppstein, *Finding the k Shortest Paths*, SIAM J. Computing, V.28(2), 652-673, 1998.

[17] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, S. Shenker, *On a Network Creation Game*, PODC 2003, 347-351.

[18] R. W. Floyd, *Algorithm 97: Shortest path*, CACM, V.5(6), 345, 1962.

[19] S. Fortune, J. Hopcroft, J. Wyllie, *The Directed Subgraph Homeomorphism Problem*, Theoretical Computer Science, V.10, 111-121, 1980.

[20] T. Groves, *Incentives in Teams*, Econometrica, V.41, 617-631, 1973.

[21] A. Hayrapetyan, É. Tardos, T. Wexler, *A Network Pricing Game for Selfish Traffic*, PODC 2005, 284-291.

[22] N. Immorlica, D. Karger, E. Nikolova, R. Sami, *First-Price Path Auctions*, EC 2005, 203-212.

[23] A. R. Karlin, D. Kempe, T. Tamir, *Beyond VCG: Frugality of Truthful Mechanisms*, FOCS 2005, 615-626.

[24] E. Koutsoupias, C. H. Papadimitriou, *Worst-Case Equilibria*, STACS 1999, 404-413.

[25] C. L. Li, S. T. McCormick, D. Simchi-Levi, *The Complexity of Finding Two Disjoint Paths with min-max Objective Function*, Discrete Applied Mathematics, V.26(1), 105-115, 1990.

[26] H. Lin, T. Roughgarden, É. Tardos, A. Walkover, *Braess's Paradox, Fibonacci Numbers, and Exponential Inapproximability*, ICALP 2005, 497-512.

[27] A. Mas-Colell, M. D. Whinston, J. R. Green, *Microeconomic Theory*, Oxford University Press, 1995.

[28] M. Mavronicolas, P. Spirakis, *The Price of Selfish Routing*, STOC 2001, 510-519.

[29] J. D. Murchland, *Braess's Paradox of Traffic Flow*, Transportation Research, V.4, 391-394, 1970.

[30] N. Nisan, A. Ronen, *Algorithmic Mechanism Design*, STOC 1999, 129-140.

[31] M. J. Osborne, A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[32] J. Oxley, *Matroid Theory*, Oxford University Press, 1992.

[33] C. H. Papadimitriou, *Algorithms, Games, and the Internet*, STOC 2001, 749-753.

[34] A. Ronen, R. Tallisman, *Towards Generic Low Payment Mechanisms for Decentralized Task Allocation – a Learning Based Approach*, IEEE Conference on E-Commerce Technology, 126-133, 2005.

[35] T. Roughgarden, *Stackelberg Scheduling Strategies*, STOC 2001, 104-113.

[36] T. Roughgarden, *Designing Networks for Selfish Users is Hard*, FOCS 2001, 472-481.

[37] T. Roughgarden, *The Price of Anarchy is Independent of the Network Topology*, STOC 2002, 428-437.

[38] T. Roughgarden, *Selfish Routing*, PhD thesis, Cornell University, Department of Computer Science, May 2002.

[39] T. Roughgarden, É. Tardos, *How Bad is Selfish Routing?*, FOCS 2000, 93-102.

[40] J. W. Suurballe, *Disjoint Paths in a Network*, Networks, V.4, 125-145, 1974.

[41] K. Talwar, *The Price of Truth: Frugality in Truthful Mechanisms*, STACS 2003, 608-619.

[42] É. Tardos, *Network Games*, STOC 2004, 341-342.

[43] A. Vetta, *Nash Equilibria in Competitive Societies, with Applications to Facility Location, Traffic Routing and Auctions*, FOCS 2002, 416-425.

[44] W. Vickrey, *Counterspeculation, Auctions and Competitive Sealed Tenders*, Journal of Finance, V.16, 8-37, 1961.

[45] R. Vohra, private communication.