Pricing of Partially Compatible Products

David Kempe*
University of Southern California
dkempe@usc.edu

Adam Meyerson
University of California, Los Angeles
awm@cs.ucla.edu

Nainesh Solanki
MediaDefender Inc.†
naineshsolanki@gmail.com

Ramnath K. Chellappa
Emory University†
ram@bus.emory.edu

ABSTRACT

In this paper, we examine a duopolistic market where the two firms compete to sell a system of components. Components are digital (firms have unlimited supply at no marginal cost), and customers are homogeneous in their component preferences. Each customer will assemble a utility maximizing system by purchasing each necessary component from one of the two firms. While components from the same firm are always compatible, pairwise compatibility of components from rival firms may vary; in addition to utility due to the quality of the system purchased, customers have negative utility for purchasing incompatible parts. We investigate algorithms and hardness results for profit-maximizing decisions of the firms with regards to their price-setting, component value-enhancing and compatibility-enabling strategies.

The users' behavior can be modeled as a minimum cut computation, and the company's strategies require addressing novel and interesting questions about graph cuts and flows. We develop a polynomial-time algorithm for finding profit-maximizing prices if the qualities and compatibilities are fixed. On the other hand, we show that finding profit-maximizing quality improvements is equivalent to the Maximum Size Bounded Capacity Cut problem, and thus NP-complete. Finally, for the problem of improving compatibilities to maximize the price, we give polynomial approximation hardness results even in very restricted cases, but show that if all components have uniform prices, and quality differences are small, then an approximation can be found in polynomial time.

*Work supported in part by NSF CAREER Award 0545855.
†Work done while the author was at the University of Southern California.

EC'07, June 13–16, 2007, San Diego, California, USA.

Categories and Subject Descriptors

K.4.4 [**Computers and Society**]: Electronic Commerce; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms

Algorithms, Economics, Theory

Keywords

Pricing, Compatibility, Duopoly, Graph, Cut

1. INTRODUCTION

For firms operating in an environment where a collection of products or components needs to work together, decisions regarding a component's compatibility with that of a rival are of strategic importance. Any price-setting behavior in such a market needs to take into account the end-consumers' disutility from purchasing partially compatible components from two or more rival firms. The complexity of decision-making in such a market can be further compounded if firms can also affect purchasing behavior by altering a component's generic value to the end-user.

Modeling such decisions has largely been the domain of research in economics [6, 7, 8, 18, 22, 23] wherein compatibility choices have been characterized as adoption of a particular standard, de jure or de facto. While this research has generally suggested that firms are better off by aligning with the leading industry standard, the results do not easily generalize to a market wherein firms produce multiple components, and there are partial incompatibilities rather than full or no compatibility.

In this paper, we address the following general version of the problem. Two companies each produce their own versions of each of a number of products. The versions produced by the two companies serve in principle the same function; for instance, each company may be producing an operating system, a word processor, a spreadsheet, and a graphics program. While the components serve the same function, they may differ in the reliability, user interface, features, efficiency, or other parameters, and thus, a user may have different valuations for the versions by the two companies. We assume that the products are *digital*; in other words, the companies have unbounded supply and there is no per-unit production cost.

In principle, components by the two companies are interchangeable; however, they will frequently not be fully compatible, and as a result, the total system valuation may be lower than the sum of individual valuations of the combined components. Thus, users will tend to use inferior products from one company in order to avoid the utility decrease due to incompatibility. The companies will be aware of those user tendencies, and exploit them in their pricing decisions and product planning. For instance, a well-known operating system sells widely for a positive price, even though it has freely available and (by most accounts) technologically superior competition. The reason is that most application software is at best partially compatible with the free competition.

In this paper, we study several aspects of this problem. We generally assume a *duopoly*, i.e., only two companies are competing. Furthermore, we assume that the market is homogenous, i.e., all consumers have identical utilities. This is a strong assumption; however, even under this assumption, the problem leads to surprising and interesting strategic decisions.

Formally, for each of n components i, and each company $\ell = 1, 2$, we are given a non-negative *valuation* $\nu_{\ell,i} \geq 0$. In addition, for each ordered pair i, i' of products, there is an *incompatibility penalty* $\gamma_{i,i'} \geq 0$, incurred by any user who combines company 1's version of component i with company 2's version of component i'. Notice that these $\gamma_{i,i'}$ will not necessarily be symmetric or binary: it is quite frequent that incompatibility manifests itself with only some lacking functionality, or additional hassle in importing file formats. After prices $p_{\ell,i}$ have been set for all components, the users will choose the system maximizing their total utility: the sum of all qualities, minus all incurred incompatibility penalties and the prices. Given these utility-maximizing decisions by the users, the company will be interested in maximizing its profit. Here, we assume that the components are digital (i.e., they exist in unlimited supply and cause no marginal cost to the companies); thus, a company's profits equal the sum of prices of all components sold. We focus on the following three questions:

1. Given knowledge of all incompatibilities and user preferences (product qualities), as well as the competitor's prices, what prices should a company choose to maximize its profit?

2. In the same scenario as above, suppose that the product prices are given, and the company has a budget B for improving the quality of its products. How should the budget be divided over the products to maximize the revenue from sales?

3. If product prices and utilities are fixed, but a company can unilaterally alter compatibilities (for instance by writing conversion tools or emulators, or by deliberately introducing novel formats or system-specific bugs), how should a company use these means to maximize its profit?

While our work focuses on the best-response strategy for a company, the scenario naturally suggests other game-theoretic questions as well, including notions of equilibria in a repeated game and the auction-theoretic view of extracting truthful valuations from customers. These topics are discussed briefly in Section 6.

1.1 Our Results

Perhaps most surprisingly, the first and most natural problem, choosing profit-maximizing best-response prices, can be solved in polynomial time. The algorithm relies on a non-trivial reduction to minimum cuts, based on insights into a novel variant of the inverse minimum cut problem.

THEOREM 1. *Profit-maximizing prices in a duopoly with identical consumers can be found in polynomial time.*

While seemingly similar, the situation is less positive for the scenario with a quality-improvement budget. We can show that the problem is equivalent, under approximation-preserving reductions, to the Maximum-Size Bounded Capacity Cut (MaxSBCC) problem defined in [15, 24]. It is thus NP-complete. Neither approximation hardness results nor non-trivial approximations are known for the MaxSBCC problem. Using techniques of Feige and Krauthgamer [9], Svitkina and Tardos give an $(O(\log^2 n), 1)$ bicriteria approximation algorithm for the MaxSBCC problem [24]. Thus, by multiplying the budget by $O(\log^2 n)$, it is possible to obtain a profit no worse than the actual optimum with the original budget.

Finally, for the problem of unilaterally changing incompatibilities, we obtain a two-fold result.

THEOREM 2.

1. *If incompatibilities can be both increased and decreased, then the problem is equivalent to* KNAPSACK. *In particular, there is a pseudo-polynomial algorithm and a PTAS.*

2. *If incompatibilities can be only decreased (i.e., products can be made more compatible, for instance by adding conversion tools or emulators), then in general, the optimum revenue is hard to approximate to within a polynomial factor $\Omega(n^{1-\epsilon})$ for each $\epsilon > 0$. However, if all prices of company 1's products are the same, and the quality differences are integers between 1 and some bound C, then a $1/C$-approximation can be obtained in polynomial time.*

The proofs for both the hardness result and approximation algorithm rely on a novel duality between special types of flows, termed S-exclusive flows, and s-t cuts S that can be turned into minimum cuts.

1.2 Related Work

The impact of incompatibilities on prices, product bundling, and profits, as well as the emergence of standards, have been studied in the past in the economics literature (see, e.g., [6, 7, 8, 18, 22, 23]). These papers assume that components are either fully compatible, adhering to a uniform standard, or fully incompatible, in which case the systems sold are entirely purchased from the same company. In addition, these papers make strong probabilistic assumptions about consumer preferences (such as assuming that users have independently uniform preferences over products), in order to derive closed-form solutions, and frequently only give solutions for two components.

Most of the models [6, 7, 18, 22] predict that the ultimate result in equilibrium would be complete compatibility, in the presence (or even absence) of positive network externalities (outside incentives such as support, resale, etc., encouraging consumers to purchase identical systems). Thus, it appears that traditional economic models are not adequate to describe the actual outcome of strategic choices in component markets with partial compatibility [13]. In particular, they do not make predictions, or provide algorithms, for determining optimal prices in any *given fixed* instance of consumer valuations.

Recently, there has been a lot of work in the theoretical computer science community on the general areas of product pricing and auctions. Several papers [1, 4, 12] have studied combinatorial optimization problems resulting from constraints on the sets of items that consumers are interested in purchasing. In addition, a lot of attention has been given to the design of incentive-compatible mechanisms [3, 17, 25], specifically in the design of profit-maximizing auctions [10, 11, 12]. In the present paper, incentive-compatibility and the elicitation of bids is not a concern, although they pose a promising direction for future research. The most relevant techniques in deriving algorithms in the setting of this paper come from the areas of unbalanced graph cuts [9, 15, 24] and inverse minimum-cut problems [2, 26].

Overview of the paper

We begin by defining the problem formally, and proving several useful basic facts, in Section 2. We then proceed with the reduction between the quality improvement problem and MaxSBCC in Section 3. Based on that reduction, we present the polynomial-time algorithm for pricing in Section 4. Section 5 discusses the algorithms and hardness results for the problem of altering compatibilities. We conclude with many open problems and future directions in Section 6.

2. PRELIMINARIES

We assume that there are n *products* (also called *components*) $i = 1, 2, \ldots, n$. We are focusing on the case of a duopoly with digital goods, i.e., each product is produced by each of two companies, at no marginal costs and with unbounded supply. Each product i by each company ℓ has an associated *valuation* $\nu_{\ell,i} \geq 0$ for the user (also called its *quality*). A *system* consists of one version of each product, i.e., purchasing each of the products $i = 1, \ldots, n$ from either company 1 or company 2. We can thus identify a system with the set $S \subseteq \{1, \ldots, n\}$ of components purchased from company 1; $\overline{S}$ is then purchased from company 2.

The total valuation of a system is determined by the individual components' valuations, but it is also affected by incompatibilities. Specifically, if a system combines two components $i \in S$ and $i' \notin S$, then the total quality of the system decreases by the *incompatibility penalty* $\gamma_{i,i'} \geq 0$. Notice that it is possible that $\gamma_{i,i'} \neq \gamma_{i',i}$, as there is no reason a priori to believe that, for instance, company 1's word processor and company 2's operating system are exactly as incompatible as company 1's operating system and company 2's word processor.

We assume that all components produced by the same company are fully compatible. This assumption is without loss of generality, so long as the company-internal incompatibilities are smaller than the ones between components of different companies. (Then, subtracting the company-internal incompatibilities does not affect the optimal solution.) The total system valuation to a user is then

$$\nu(S) = \sum_{i \in S} \nu_{1,i} + \sum_{i \notin S} \nu_{2,i} - \sum_{i \in S, i' \notin S} \gamma_{i,i'}.$$

If company 1 charges price $p_{1,i} \geq 0$ for component i, and company 2 charges $p_{2,i}$, then we assume that the users choose the system S maximizing the *utility*

$$U(S) = \nu(S) - \sum_{i \in S} p_{1,i} - \sum_{i \notin S} p_{2,i}.$$

The system S maximizing $U(S)$ can be characterized in terms of the minimum cut of a graph. Define a graph G with node set $V = \{s, t, v_1, \ldots, v_n\}$, and directed edges (s, v_i) of capacity $\nu_{1,i} + p_{2,i}$ and (v_i, t) of capacity $\nu_{2,i} + p_{1,i}$, as well as directed edges (v_i, v_j) of capacity $\gamma_{i,j}$. Table 1 gives example qualities, prices, and incompatibilities for a three-component system, and Figure 1 shows the corresponding graph.

Component	$\nu_{1,i}$	$\nu_{2,i}$	$p_{1,i}$	$p_{2,i}$
Operating System	2	4	1	0
Spreadsheet	3	1	2	2
Word Processor	4	3	2	1

↓ Comp. 1 | Comp. 2 →	OS	SS	WP
Operating System		0	2
Spreadsheet	3		0
Word Processor	1	1	

Table 1: Example prices and incompatibilities

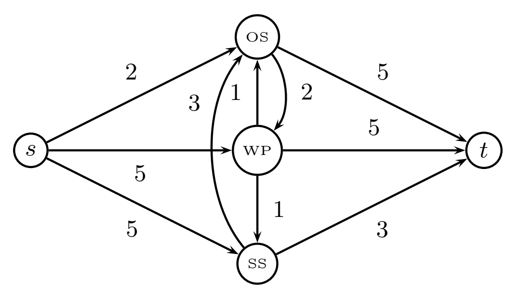

Figure 1: The corresponding graph G

LEMMA 3. *A system S maximizes $U(S)$ if and only if $(S + s, \overline{S} + t)$ is a minimum cut in G.*

Proof. The capacity of any cut $(S+s, \overline{S}+t)$ in G is exactly

$$\begin{aligned} &\sum_{i \in S} (\nu_{2,i} + p_{1,i}) + \sum_{i \notin S} (\nu_{1,i} + p_{2,i}) + \sum_{i \in S, i' \notin S} \gamma_{i,i'} \\ &= \sum_i (\nu_{1,i} + \nu_{2,i}) - \Big(\sum_{i \in S} \nu_{1,i} + \sum_{i \notin S} \nu_{2,i} \\ &\quad - \sum_{i \in S, i' \notin S} \gamma_{i,i'} - \sum_{i \in S} p_{1,i} - \sum_{i \notin S} p_{2,i} \Big) \\ &= \sum_i (\nu_{1,i} + \nu_{2,i}) - U(S). \end{aligned}$$

Hence, maximizing $U(S)$ is equivalent to minimizing the capacity of $(S + s, \overline{S} + t)$. ∎

Remark 4. *If we were to study competition between more than two companies, then the resulting optimization problem for a user would be to assign labels (or colors) to the nodes of a graph, where edge-specific penalties are incurred for labeling adjacent nodes with different colors. The resulting optimization problem for the user is NP-complete for any fixed number $L \geq 3$ of companies. It is akin to the problem of metric labeling studied in [5, 19].*

The above formulation of course immediately implies an algorithm for a consumer to choose an optimal system: compute the minimum s-t cut of G, and purchase the components on the s-side from company 1, and those on the t-side from company 2.

Using the cut-based characterization, we can thus reformulate the optimization problem in terms of changing edge costs so as to maximize the total value of nodes on the s side of the cut. Recall that we are focusing on the case where company 2 has already determined all its prices and product qualities, and company 1 is seeking a best-response strategy. (The questions of repeated, alternating iterations of this game, equilibria, or dominating strategies, are briefly discussed in Section 6.)

1. The *budgeted improvement problem* consists of increasing the capacity of edges out of s by a total of at most B so as to maximize the total price $p_{1,S}$ of products on the s-side of the resulting minimum cut.

2. The *pricing problem* asks to assign prices $p_{1,i}$ to nodes so that in the minimum cut with edge capacities (v_i, t) determined by the $p_{1,i}$ values, the total price of components on the s-side is maximized.

3. The *compatibility alteration problem* asks to change (or decrease, in the case of solely improving compatibility) the capacities of edges not incident with s or t so as to maximize the total price $p_{1,S}$ of components on the s-side of the minimum cut.

The problem thus bears some similarity with the inverse minimum cut problem [2, 26], in which the edge capacities of a given graph G are to be altered as little as possible (measured as the sum of changes over all edges) so as to ensure that a given cut $(S, \overline{S})$ becomes the minimum s-t cut. Notice, though, that our formulation differs in two critical aspects: we only allow certain limited changes, and the goal is not to obtain one particular cut, but to maximize the total price of components on the s-side.

3. BUDGETED IMPROVEMENT PROBLEM

We first consider the problem of improving product valuations with a limited budget, so as to maximize the total revenue of products sold. As argued above, this is equivalent to being given a graph with non-negative edge capacities c_e, as well as node prices $p_i := p_{1,i}$. The goal is to add a total of at most B units of capacity to edges leaving the source s, so as to ensure that in the resulting graph, the total price p_S for the s-side S of the minimum s-t cut is maximized.

In deriving an equivalent formulation, the following lemma is crucial, relating the necessary capacity increase to the difference in cut capacities. To the best of our knowledge, this lemma has not appeared elsewhere.

Lemma 5. *Let $G = (V, E)$ be a graph with capacities c_e, and let $(S^*, \overline{S^*})$ be a minimum s-t cut of G, of capacity C^*. Let $S \supseteq S^*$ define an s-t cut $(S, \overline{S})$ of capacity $C \geq C^*$. Then, by increasing the edge capacities out of s by a total of at most $C - C^*$, it is possible to ensure that there is a minimum cut $(S', \overline{S'})$ with respect to the new capacities, with $S' \supseteq S$.*

In words, this lemma is saying that in order to ensure that at least all of S ends up on the s-side of a minimum cut with altered capacities, it is enough to increase capacities out of s by a total of at most the difference between the capacities of $(S, \overline{S})$ and the original minimum cut capacity.

Proof. Consider adding a new, parallel, edge $e_i = (s, i)$ for each node $i \in S \setminus S^*$, with capacity $c_{e_i} = C - C^*$. In the new graph, the capacity of each s-t cut $(A, \overline{A})$ with $S \not\subseteq A$ must be at least C, while the capacity of each cut $(A, \overline{A})$ with $S \subseteq A$ stays unchanged. In particular, the new minimum s-t cut must be some $(S', \overline{S'})$ with $S' \supseteq S$.

Because $(S, \overline{S})$ is a cut of capacity C, the maximum flow f in the new graph can have value at most C. Also, because the previous maximum flow f^* had value C^*, the new flow f can be obtained from f^* by adding an augmenting flow $f - f^*$ of value $C - C^*$. In particular, the total flow f on the newly added edges e_i is at most $C - C^*$. By reducing the capacities of the newly added edges e_i to the flow $c'_{e_i} := f_{e_i}$, f will certainly remain a maximum flow, and hence, $(S, \overline{S})$ will remain a minimum cut. At the same time, the total amount of capacity added is $\sum_i c'_{e_i} = \sum_i f_{e_i} \leq C - C^*$. This completes the proof of the lemma. ■

Using this lemma, it becomes rather straightforward to characterize the maximum profit that can be obtained by increasing the edge capacities out of s by a total of at most B. For a particular set S of products (or a superset of it) can be sold by company 1 if and only if the capacity of the cut $(S+s, \overline{S}+t)$ exceeds that of the minimum cut by at most B. If C^* denotes the minimum cut's capacity, we are thus seeking the set S maximizing p_S, subject to the constraint that the capacity of $(S, \overline{S})$ be at most $C^* + B$. This problem is exactly the weighted Maximum-Size Bounded Capacity Cut (MaxSBCC) problem introduced and studied in [15, 24]. It can be written as an integer linear program as follows (this formulation will be useful in Section 4):

$$\begin{array}{lll} \text{Maximize} & \sum_i p_i \cdot x_i & \\ \text{subject to} & x_s = 1, x_t = 0 & \\ & y_e \geq x_i - x_j & \text{for each } e = (i,j) \\ & \sum_{e \in E} y_e \cdot c_e \leq C^* + B & \\ & x_i, y_e \in \{0, 1\}. & \end{array} \quad (1)$$

Here, $x_i = 1$ corresponds to a node being on the s-side of the cut, and $y_e = 1$ corresponds to edge e being cut. While it is known [15, 24] that MaxSBCC is NP-hard, even for unit prices, neither an approximation hardness result nor non-trivial approximation algorithms are known at this point.

Using the techniques of Feige and Krauthgamer [9], it is possible [24] to give an $(O(\log^2 n), 1)$ bicriteria approximation algorithm. Thus, by increasing the capacity increase budget by a factor of $O(\log^2 n)$, one can obtain a profit no worse than the best possible with the original budget. However, single-criteria results are not known at this point.

4. PRICING PROBLEM

We can use the insights from the previous section to derive a polynomial-time algorithm for determining the product prices giving highest total profit. Imagine setting very high prices p_i for all products i, so high that the minimum cut separates s from the rest of the graph, i.e., company 1 does not sell any products. Now suppose that the prices are reduced by a total "rebate" of at most R. A reduction in price is equivalent to an improvement in quality, hence we can think of improving the quality by at most R. At the same time, the rebate will decrease the profit by R, so the new optimization problem becomes

$$\begin{array}{lll} \text{Maximize} & \sum_i p_i \cdot x_i - R & \\ \text{subject to} & x_s = 1, x_t = 0 & \\ & y_e \geq x_i - x_j & \text{for each } e = (i,j) \\ & \sum_{e \in E} y_e \cdot c_e \leq C^* + R & \\ & x_i, y_e \in \{0,1\}. & \end{array} \qquad (2)$$

At optimality, the "improvement" constraint $\sum_{e \in E} y_e \cdot c_e \leq C^* + R$ must be tight (else R could be decreased, for a better solution), so the maximization problem is equivalent to

$$\begin{array}{lll} \text{Maximize} & \sum_i p_i \cdot x_i + C^* - \sum_{e \in E} y_e \cdot c_e & \\ \text{subject to} & x_s = 1, x_t = 0 & \\ & y_e \geq x_i - x_j & \text{for each } e = (i,j) \\ & x_i, y_e \in \{0,1\}. & \end{array} \qquad (3)$$

The optimization problem is thus to find an s-t cut $(S, \overline{S})$ maximizing $\sum_{i \in S} p_i - \sum_{i \in S, j \notin S} c_{(i,j)}$, or equivalently — since C^* is a constant — minimizing $\sum_{i \notin S} p_i + \sum_{i \in S, j \notin S} c_{(i,j)}$. But this is exactly the minimum cut in the graph G if we increase the capacity of all edges (s, i) out of s by p_i, or, equivalently, decrease the capacity of all edges (i, t) into t by p_i. That minimum cut is exactly the optimum system if all of company 1's prices are 0. Thus, we have proved the following lemma:

LEMMA 6. *With the optimal price setting, company 1 sells exactly the same set S^* of products as if it gives away all its products for free.*

Once the optimal sold set S^* has been determined, we still need to assign the corresponding prices. One way would be to set up an exponential-sized linear program, maximizing the sum of prices on the nodes in S^*, subject to the constraint that the capacity of each cut contained in S^* be at least the capacity of the cut $(S^*, \overline{S^*})$. While this LP has exponential size, it has a polynomial-time separation oracle (via a minimum cut computation), and can thus be solved in polynomial time.

A more efficient approach uses a single Max-Flow computation, following the approach of Lemma 5. First, in the graph with product prices 0, find a maximum s-t flow. Thus, the cut $(S^*, \overline{S^*})$ will be a minimum cut. Now, add new (parallel) edges $e_i = (i, t)$ for $i \in S^*, i \neq s$, with some large capacity, and find a maximum augmenting flow f' from s to t. If we change the capacity of each newly added edge e_i to f'_{e_i}, then $(S^*, \overline{S^*})$ is a saturated cut, and thus a minimum cut for the flow $f + f'$. On the other hand, because f' is a maximum augmenting flow, it is impossible to increase the total capacities of the e_i for $i \in S^*$ beyond the value of f' without having $(S^*, \overline{S^*})$ cease to be a minimum cut. Hence, the optimum profit is obtained by assigning each product in S^* a price of f'_{e_v}, and each product not in S^* an arbitrary price. We have thus proved:

THEOREM 7. *There is a polynomial-time algorithm for selecting profit-maximizing best-response prices in a duopoly with identical consumers.*

5. COMPATIBILITY ALTERATION

In this section, we discuss results regarding deliberate and unilateral changes in product compatibilities. We assume that all prices and costs, and thus the capacities of all edges incident with the source or sink, are fixed. The goal is to change capacities of edges $e = (i, j)$ with neither i nor j being the source or sink, so as to maximize the total price of nodes on the s-side S of the minimum cut in the resulting graph. We first discuss the problem in which compatibilities can be arbitrarily increased or decreased, and then the one where products can only be made more compatible, i.e., edge capacities can only be lowered.

5.1 Arbitrary Capacity Changes

If the capacities c_e can be increased or decreased, then the original capacities may be assumed to be 0 without loss of generality. We claim that the problem is exactly identical with the KNAPSACK problem.

PROPOSITION 8. *Without loss of generality, the optimum solution uses only capacities 0 or ∞ for edges not incident with s or t.*

Proof. If $(S, \overline{S})$ is a minimum cut with respect to capacities c, then it is also a minimum cut if we change the capacities of all edges $e \in (S \setminus \{s\}) \times (\overline{S} \setminus \{t\})$ to 0, and the capacities of all edges $e \in \overline{S} \times S$, $e \in S \setminus \{s\}$ and $e \in \overline{S} \setminus \{t\}$ to ∞. ∎

We first observe that the customer can always be forced to buy products from only one company, by setting all incompatibilities to infinity. If company 1's products have a total valuation minus price higher than those of the competitor, then this will be the optimum solution and the problem is solved. Assuming this is not the case, a set S can be the set of products sold if and only if $(S, \overline{S})$ is a minimum cut under the capacity assignment described in Proposition 8. Under this capacity assignment, the only cuts with finite capacity are $(S, \overline{S}), (\{s\}, V \setminus \{s\}), (V \setminus \{t\}, \{t\})$. The capacity of the last two cuts represent the total valuation minus price of company 1's products and the competitor's products, respectively, and we are assuming that company 2's products have larger total valuation minus price.

Thus, S can be the set of products sold if and only if the capacity of $(S, \overline{S})$ is smaller than that of $(\{s\}, \{t, 1, \ldots, n\})$, i.e., if

$$\sum_{i \in S} (\nu_{2,i} + p_{1,i}) + \sum_{i \notin S} (\nu_{1,i} + p_{2,i}) \quad \leq \quad \sum_i (\nu_{1,i} + p_{2,i}),$$

or, equivalently,

$$\sum_{i \in S} (\nu_{1,i} + p_{2,i} - (\nu_{2,i} + p_{1,i})) \quad \geq \quad 0.$$

The optimum solution S will always contain all i such that $d(i) := \nu_{1,i} + p_{2,i} - (\nu_{2,i} + p_{1,i}) \geq 0$, and we can write $D := \sum_{i:d(i) \geq 0} d(i)$. Then, the optimum solution S

is the one maximizing p_S subject to the constraint that $\sum_{i \in S, d(i)<0}(-d(i)) \leq D$. This is exactly a KNAPSACK problem with values p_i, "weights" $-d(i)$, and weight bound D. The reduction works both ways. As a result, we obtain a pseudo-polynomial time algorithm and PTAS for the problem of arbitrarily setting compatibilities to maximize profit.

5.2 Improving Compatibilities

When company 1 can only improve compatibilities unilaterally, but not worsen them, then we are looking for new capacities $c'_e \leq c_e$ on each edge e not incident with the source or sink, such that p_S is maximized, where S is the minimum cut with respect to the capacities c'_e. We will call capacities c' satisfying $c'_e \leq c_e$ on all edges e, and $c'_e = c_e$ for all e incident with the source or sink *valid capacities*. If S is the minimum s-t cut for some valid capacities, we call S a *minimizable cut*.

In the further description, we will assume without loss of generality that no node i has both an edge from s, and an edge into t. If both (s, i) and (i, t) were edges, then we could remove the edge with smaller capacity, and replace the other capacity by the difference. It is then easy to see that we have simply reduced the capacity of *every* cut in the graph by the same constant, and not affected the problem otherwise. Intuitively, the capacity of the edge (s, i) or (i, t) is then the *utility difference* between the versions of product i produced by companies 1 and 2: how much better (or worse) is the version of company 1, taking into consideration the price? We will sometimes refer to the edge capacity by this name. Since we will frequently refer to edges neither of whose endpoints is s or t, we will call such edges *internal edges*.

Similar to Proposition 8, we now prove that we can restrict our attention to specific types of valid capacities.

PROPOSITION 9. *$(S, \overline{S})$ is a minimizable cut if and only if it is a minimum cut with respect to the capacities $c'_e = 0$ for $e \in S \times \overline{S}$, and $c'_e = c_e$ for all other edges e.*

Proof. Suppose $(S, \overline{S})$ is a minimizable cut, say, for capacities c''. For some edge e not crossing $(S, \overline{S})$, change the capacity to $c'_e := c_e$. This can only increase the costs of other cuts in the graph; however, it does not increase the cost of the cut $(S, \overline{S})$. Therefore, the cut $(S, \overline{S})$ remains minimum. Now consider any edge e crossing $(S, \overline{S})$, with neither endpoint at t. If we reduce the capacity of this edge by ϵ, then the capacity of the cut $(S, \overline{S})$ decreases by ϵ, and the capacity of any other cut decreases by at most ϵ. Again, $(S, \overline{S})$ remains minimum. By repeating this process for each edge, the theorem is proved. ■

We next show a duality between minimizable cuts and flows f we term S-exclusive: We say that f is *S-exclusive* for a set S containing the source, but not the sink, if $f_{(i,t)} = c_{(i,t)}$ for all $i \in S$, and $f_{(i,j)} = 0$ for all $j \notin S, j \neq t$.

THEOREM 10. *Let $(S, \overline{S})$ be a partition such that $c_{(s,j)} = 0$ for all $j \notin S$. Then, $(S, \overline{S})$ is minimizable if and only if there exists an S-exclusive s-t flow f.*

Proof. If $(S, \overline{S})$ is minimizable, then we consider the capacity function $c'_e = 0$ for all internal edges e crossing $(S, \overline{S})$, and $c'_e = c_e$ for all other edges. Then, $(S, \overline{S})$ is a minimum cut for c', and has capacity exactly $\sum_{i \in S} c_{(i,t)}$. Thus, there is a flow f of that value with respect to the capacities c'. But then, f cannot use any edges (i, j) with $i \in S, j \notin S, j \neq t$, and must therefore saturate all edges (i, t) for $i \in S$. Thus, f is S-exclusive, and it is of course also a feasible flow with respect to c.

On the other hand, if f is an S-exclusive flow, then it still remains feasible if we define capacities c'_e as above. As the value of f is $\sum_{i \in S} c_{(i,t)}$, which is also the capacity of the cut $(S, \overline{S})$ under c', we know that $(S, \overline{S})$ must be a minimum cut under c', and in particular is minimizable. ■

As an immediate corollary of this theorem, we can show the following:

COROLLARY 11. *If all edges into t have capacity 0 or 1, and the prices $p_{1,i}$ are uniform (without loss of generality, $p_{1,i} = 1$ for all i), then the best minimizable cut $(S, \overline{S})$ can be found in polynomial time.*

Proof. We start by computing an (integral) maximum s-t flow f. Then, we can modify f so that if node i has positive incoming flow, and an edge of capacity 1 to the sink t, then $f_{(i,t)} = 1$ (simply reroute one unit of flow leaving from i to go directly to t). Letting S be the set of nodes i with $f_{(i,t)} = c_{(i,t)}$, we see that f is S-exclusive, and hence S is minimizable. The size of S is equal to the number of nodes with $c_{(i,t)} = 0$ plus the value of f, and because f is a maximum flow, no larger minimizable set S exists. ■

The algorithm from the proof of Corollary 11 can be generalized to give a $1/C$-approximation algorithm if all prices are uniform (without loss of generality, $p_{1,i} = 1$ for all i), and all edges into t have integer capacities from $\{1, \ldots, C\}$.[1] The generalization of the algorithm is presented here as Algorithm 1:

Algorithm 1 Greedy Augmentation Algorithm

```
1: Start with S := {i | c_(i,t) = 0}.
2: repeat
3:    Among all i ∉ S such that S ∪ {i} is minimizable, let
      i be the one minimizing c_(i,t).
4:    Add i to S.
5: until S ∪ {i} is not minimizable for any i.
6: Return the set S.
```

The initial set S is always minimizable, because it contains only those components for which company 1's version is superior — by making them fully compatible, company 1 can ensure to sell all of those components. In each iteration, minimizability can be checked in polynomial time via a single minimum-cut computation.

THEOREM 12. *If $C = \max_{i \in V} c_{(i,t)}$, then the set $\hat{S}$ found by the greedy algorithm satisfies $|\hat{S}| \geq \frac{1}{C} \cdot |S^*|$, where S^* is the largest minimizable set.*

Proof. Because both $\hat{S}$ and S^* are minimizable, there exist corresponding exclusive flows $\hat{f}$ and f^*. For any set S,

[1]While this case appears fairly restrictive, it does apply to software component markets where individual components are priced (roughly) the same, and the valuation differences are not terribly large. The results are also essentially best possible in light of the approximation hardness results presented below.

not necessarily minimizable, we define $\phi(S)$ to be the maximum value of any s-t flow that does not use any internal edges (i,j) with $i \in S, j \notin S$. In particular, by Theorem 10, for any minimizable set S, we have that $\phi(S) = \sum_{i \in S} c_{(i,t)}$, and for all non-minimizable sets S, $\phi(S) < \sum_{i \in S} c_{(i,t)}$.

Let A be a non-empty node set disjoint from $\hat{S}$. We prove the following bound:

$$\phi(\hat{S} \cup A) \quad \leq \quad \phi(\hat{S}) + \sum_{i \in A} c_{(i,t)} - |A|. \qquad (4)$$

Assume that this inequality were not true, and let A be a smallest set violating it, of size k. Let f be an s-t flow of value $\phi(\hat{S} \cup A)$, not using any edges out of $\hat{S} \cup A$ except edges to t. We observe that such a flow can be computed for $\hat{S} \cup A$ by simply removing the internal edges out of $\hat{S} \cup A$ from the graph, and augmenting the flow for $\hat{S}$. Thus, it follows that some such flow saturates all the edges from nodes in $\hat{S}$ to t.

Consider the (non-empty) set $\mathcal{F}$ of all integral and acyclic such flows f. For each flow $f \in \mathcal{F}$, we can define the graph G_f on the set A that contains an edge e if and only if $f_e > 0$. Because each such G_f is acyclic, it can be topologically sorted. For any valid topological sort $\sigma : A \to \{1, \ldots, |A|\}$ of G_f, we define a potential function $\Phi(f, \sigma) := \sum_{i \in A} \sigma(i) \cdot f_{(i,t)}$. Fix f and σ to be a pair minimizing $\Phi(f, \sigma)$.

Now, by the Pigeon Hole Principle, because we assumed that $\phi(\hat{S} \cup A) \geq \phi(\hat{S}) + \sum_{i \in A} c_{(i,t)} - |A| + 1$, there must be at least one node $i \in A$ such that $f_{(i,t)} = c_{(i,t)}$. Among all such nodes, let i be the one with smallest index $\sigma(i)$. Then, we claim that all the flow into i must be from nodes $k \in \hat{S}$. For $c_{s,i} = 0$ for all $i \notin \hat{S}$ by the starting condition of the algorithm, and if at least one unit of flow came from some node $j \in A$, then j would have to have a smaller index $\sigma(j) < \sigma(i)$. By minimality of the index of i, the node j cannot saturate its edge to t. Thus, we could change f, by rerouting the one unit that is currently going from j to i to t, and making it go directly from j to t. But this would give a new flow f', and since the same ordering σ is still a correct topological sort of $G_{f'}$, the pair (f', σ) would have a strictly lower potential function, a contradiction.

Now, given that all of the flow into i is along edges across the cut $(\hat{S}, \overline{S})$, we can define a new flow f' by starting from f, and setting $f'_{(k,j)} = 0$ for all $k \in \hat{S}, j \in A, j \neq i$. The resulting flow is $(\hat{S} \cup \{i\})$-exclusive, proving that $(\hat{S} \cup \{i\})$ is minimizable. But then, the greedy algorithm would not have terminated, so we again obtain a contradiction. Thus, we have proved inequality (4).

Finally, we apply inequality (4) to the set $A = S^* \setminus \hat{S}$. Because $\hat{S} \cup A \supseteq S^*$, the monotonicity of ϕ implies that $\phi(\hat{S}) + \sum_{i \in S^* \setminus \hat{S}} c_{(i,t)} - |S^* \setminus \hat{S}| \geq \phi(S^*)$. Since both S^* and $\hat{S}$ are minimizable, we have $\phi(S^*) = \sum_{i \in S^*} c_{(i,t)}$, and similarly for $\hat{S}$. Now, we can rearrange to obtain $\sum_{i \in \hat{S} \setminus S^*} c_{(i,t)} \geq |S^* \setminus \hat{S}|$. Because the maximum capacity is C, we can bound the left-hand side to be at most $C|\hat{S}| - C|S^* \cap \hat{S}|$, and using that $C \geq 1$, we obtain that $|\hat{S}| \geq |S^*|/C$, completing the proof. ■

Unfortunately, the above approximation result is essentially the best one can hope for. Even if all prices $p_{1,i}$, and all capacities of edges (i,t), are either 0 or 1, the maximum revenue cannot be approximated to better than $n^{1-\epsilon}$ for any $\epsilon > 0$ unless NP=ZPP. Similarly, even if the prices are uniform, if the edges costs for edges (i,t) can be arbitrary, the maximum revenue (size of the s-side S) cannot be approximated to better than $n^{1/3-\epsilon}$ unless NP=ZPP. Thus, there is not much room to improve the approximation guarantee from Theorem 12.

THEOREM 13. *Both of the following approximation hardness results hold unless NP=ZPP:*

1. *Even if all prices $p_{1,i}$ and edge capacities $c_{(i,t)}$ into the sink are either 0 or 1, the maximum total price p_S for a minimizable cut $(S, \overline{S})$ cannot be approximated to within $O(n^{1-\epsilon})$, for any $\epsilon > 0$.*

2. *Even if all prices $p_{1,i}$ are 1, the maximum size of $|S|$ for a minimizable cut $(S, \overline{S})$ cannot be approximated to within $O(n^{1/3-\epsilon})$, for any $\epsilon > 0$.*

Proof. Both hardness results are by (nearly) approximation-preserving reductions from the INDEPENDENT SET problem, which is hard to approximate to within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ unless NP=ZPP [14].

1. Given a graph $G = (V, E)$ with n vertices and m edges, we define a new graph $G' = (V', E')$ with a vertex u_e for each edge $e \in E$, $d(v)$ vertices $w_{v,1}, \ldots, w_{v,d(v)}$ for each $v \in V$ (where $d(v)$ is the degree of v in G), as well as a source s and sink t. For each edge $e = (v, v') \in E$, we have an edge of capacity 1 from s to u_e, and edges of capacity 1 from u_e to both $w_{v,1}$ and $w_{v',1}$. Each $w_{v,i}$ has an edge of capacity 1 to t, and an edge of capacity $d(v)$ to $w_{v,i+1}$ The nodes $w_{v,d(v)}$ have price 1, and all other nodes have price 0. Figure 2 shows an example of a graph obtained by this reduction.

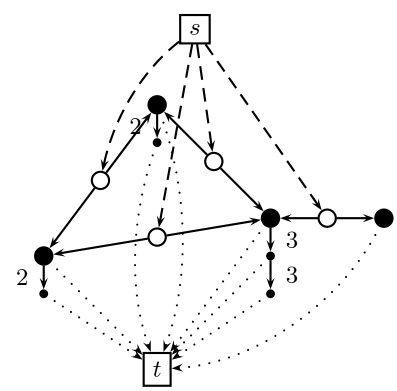

Figure 2: The graph G' obtained by a reduction, starting from a triangle with an edge added. The large solid nodes are the w_{v1}, corresponding to the original nodes v. Small solid nodes are the w_{vi} for $i > 1$. Nodes u_e are depicted with empty circles, and correspond to the edges they lie on. All arcs have capacity 1, unless labeled otherwise. Dashed and dotted arcs only serve to improve legibility.

If S is an independent set in G, then define S' to be all nodes u_e (for all edges e), as well as all nodes $w_{v,i}$ for $v \in S$. Define a flow f in G' as follows: for each edge e with an endpoint v in S, route one unit of flow from s to u_e to $w_{v,0}$. Notice that because S is independent, v is uniquely determined. In this way, each node $w_{v,0}$ for $v \in S$ has $d(v)$ units of flow arriving. These are then routed to t in the only possible way, by saturating all edges from the $w_{v,i}$ to t, and routing the remaining flow from $w_{v,i}$ to $w_{v,i+1}$. Thus, we obtain an S'-exclusive flow, and by Theorem 10, S' is minimizable. The total price of S' is exactly $|S|$, because exactly the vertices $w_{v,d(v)}$ contribute one unit of price each.

Conversely, consider a minimizable cut $(S', \overline{S'})$ in G'. Thus, there is an S'-exclusive flow f in G'. Because the only edge into $w_{v,i+1}$ is from $w_{v,i}$, for any i, we know that if $w_{v,i+1} \in S'$, then $w_{v,i} \in S'$ also. (Otherwise, f could not be S'-exclusive.) Assume for contradiction that for two distinct vertices v, v' sharing an edge $e = (v, v') \in E$, we have $w_{v,d(v)} \in S'$ and $w_{v',d(v')} \in S'$. Because the edges from both of these vertices and from all $w_{v,i}$ and $w_{v',i}$ to t must be saturated by f, the total flow into them must be at least $d(v) + d(v')$. But the total number of distinct edges incident with v or v' in G is at most $d(v) + d(v') - 1$, because they share an edge. The only flow into $w_{v,0}$ and $w_{v',0}$ can come from nodes u_e, so the total flow into them is at most $d(v) + d(v') - 1$, contradicting an outgoing flow of at least $d(v) + d(v')$. Thus, the set S of nodes v with $w_{v,d(v)} \in S'$ forms an independent set. And the size of S is exactly the total price of S'. Thus, we have an exactly approximation preserving reduction from INDEPENDENT SET.

2. For the second part of the theorem, we can use a nearly identical reduction. Instead of $d(v)$ vertices $w_{v,i}$, we have $m + 1$ vertices $w_{v,1}, \ldots, w_{v,m+1}$ for each v. The capacity of each edge from $w_{v,i}$ to $w_{v,i+1}$ is now 1, the edge from $w_{v,1}$ to t has capacity $d(v) - \frac{1}{4}$, and each edge from $w_{v,i}$ to t has capacity $\frac{1}{4m}$ for $i > 1$. The remaining nodes and edges, and their capacities, stay the same as in the first reduction.

   By a nearly identical argument as in the first proof, S' is minimizable if and only if (1) $S := \{v \mid w_{v,0} \in S'\}$ is independent, (2) for each i, if $w_{v,i+1} \in S'$, then $w_{v,i} \in S'$, and (3) S' includes the node u_e for each edge e incident with a node in S. Therefore, for each independent set S, the set $\{u_e \mid e \in E\} \cup \{w_{v,i} \mid v \in S\}$ defines a minimizable cut of size $m + (m + 1) \cdot |S|$, and conversely, for each minimizable set S', the set S defined above is an independent set of size at least $(|S'| - m)/(m + 1) \geq \frac{|S'|}{m+1} - 1$. Thus, the maximum sizes of S and S' are (up to an additive 1) exactly related by the constant factor $m+1$, and the approximation hardness of INDEPENDENT SET implies an approximation hardness of $O(n^{1/3-\epsilon})$, because the number of nodes in G' is $O(n^3)$. ∎

6. CONCLUSIONS

In this paper, we introduced the problem of product pricing in a duopoly under partial compatibilities. We showed that even if all consumers have identical valuations, incompatibilities between products lead to intricate pricing questions. Perhaps most surprisingly, the revenue-maximizing prices can be found in polynomial time. For a variant in which product qualities can be improved with a given budget, we showed the equivalence of our problem with the Maximum-Size Bounded Capacity Cut problem. For the variant in which product compatibilities can be unilaterally changed, we showed an equivalence to KNAPSACK, and for the variant where compatibilities can only be improved, we used an interesting duality between minimizable cuts and exclusive flows to derive approximation hardness results and approximation guarantees.

This paper suggests a number of interesting directions for future research. Perhaps most importantly, one should study the effect of different users' valuations on pricing decisions. Can (approximately) optimal prices be obtained in polynomial time for arbitrary valuations? Does the problem become easier if one assumes that all users have the same valuations, up to small random perturbations?

An equally natural generalization is to consider more than two companies. Even the problem of finding the best system (from a user's perspective) becomes NP-hard in that case, and it seems likely that this will also impact the guarantees that can be obtained for pricing and related problems. In fact, it is not clear that the product pricing problem in that case should necessarily be in NP.

From a practical perspective, companies will likely not rely on any one of the techniques for revenue maximization; rather, they will devise a combined strategy involving changes in price, product quality, and compatibilities. It should not be too difficult to define a natural notion of combining the different decisions.

The game-theoretic aspects of the proposed problem also offer many exciting future research questions. If the change of prices, compatibilities, and qualities is regarded not as a best-response question for one company, but as a game played by multiple companies, what can be said about the outcome of this game? It is well known (see, e.g., [16]) that first-price auctions frequently do not have Nash Equilibria, and the game contains first-price auctions as a special case, where the competing companies are the bidders. Thus, Nash Equilibria in the strict sense will not be applicable as a solution concept. Instead, it would be interesting to investigate which notions of the outcome of a game could be applied in this context, and whether the model allows us to draw interesting conclusions about the emergence of single or multiple standards, or pricing anomalies.

Finally, one can also regard the pricing decisions as a game between the companies and the potential consumers. As in many pricing and auction scenarios [20, 21], consumers may have an incentive to misrepresent their true valuations to encourage companies to lower their prices. In the spirit of previous work [10, 11, 12], it then becomes interesting to devise approximately competitive auction mechanisms extracting truthful valuations from the customers.

Acknowledgments

We would like to thank Ravinder Reddy Dharmapuram for useful discussions, and anonymous referees for helpful comments.

7. REFERENCES

[1] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for multi-product pricing. In *Proc. 31st Intl. Colloq. on Automata, Languages and Programming*, 2004.

[2] R. Ahuja and J. Orlin. Inverse optimization, Part I: Linear programming and general problem. *Operations Research*, 35:771–783, 2001.

[3] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, 2001.

[4] P. Briest and P. Krysta. Single-minded unlimited supply pricing on sparse instances. In *Proc. 17th ACM Symp. on Discrete Algorithms*, 2006.

[5] C. Chekuri, S. Khanna, S. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proc. 12th ACM Symp. on Discrete Algorithms*, 2001.

[6] N. Economides. Desirability of compatibility in the absence of network externalities. *American Economic Review*, 79:1165–1181, 1989.

[7] J. Farrell and G. Saloner. Standardization, compatibility, and innovation. *Rand Journal of Economics*, 16:70–83, 1985.

[8] J. Farrell and G. Saloner. Converters, compatibility, and control of interfaces. *Journal of Industrial Economics*, 40:9–36, 1992.

[9] U. Feige and R. Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM J. on Computing*, 31:1090–1118, 2002.

[10] A. Fiat, A. Goldberg, J. Hartline, and A. Karlin. Competitive generalized auctions. In *Proc. 34th ACM Symp. on Theory of Computing*. ACM Press, 2002.

[11] A. Goldberg, J. Hartline, and A. Wright. Competitive auctions and digital goods. In *Proc. 12th ACM Symp. on Discrete Algorithms*, pages 735–744. ACM/SIAM, 2001.

[12] V. Guruswami, J. Hartline, A. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *Proc. 16th ACM Symp. on Discrete Algorithms*, pages 1164–1173, 2005.

[13] O. Hanseth. *Information Technology as Infrastructure*. PhD thesis, Gothenburg University, 1996.

[14] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[15] A. Hayrapetyan, D. Kempe, M. Pál, and Z. Svitkina. Unbalanced graph cuts. In *Proc. 13th European Symp. on Algorithms*, 2005.

[16] N. Immorlica, D. Karger, E. Nikolova, and R. Sami. First-price path auctions. In *Proc. 7th ACM Conf. on Electronic Commerce*, 2005.

[17] A. Karlin, D. Kempe, and T. Tamir. Beyond VCG: Frugality of truthful mechanisms. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, 2005.

[18] M. Katz and C. Shapiro. Network externalities, competition, and compatibility. *American Economic Review*, 73:424–440, 1985.

[19] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric partitioning and markov random fields. In *Proc. 40th IEEE Symp. on Foundations of Computer Science*, 1999.

[20] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13:227–286, 1999.

[21] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proc. 44th IEEE Symp. on Foundations of Computer Science*, 2003.

[22] C. Matutes and P. Regibeau. "Mix and Match": Product compatibility without network externalities. *The RAND Journal of Economics*, 19:221–234, 1988.

[23] C. Matutes and P. Regibeau. Compatibility and bundling of complementary goods in a duopoly. *The Journal of Industrial Economics*, 40:37–54, 1992.

[24] Z. Svitkina and E. Tardos. Min-max multiway cut. In *Proc. 7th Intl. Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 2004.

[25] K. Talwar. The price of truth: Frugality in truthful mechanisms. In *Proc. 21st Annual Symp. on Theoretical Aspects of Computer Science*, 2003.

[26] J. Zhang and M. Cai. Inverse problem of minimum cuts. *ZOR Mathematical Methods of Operations Research*, 48:51–88, 1998.