Computational Aspects of Combinatorial Pricing Problems

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Dortmund
am Fachbereich Informatik
von

Patrick Briest

Dortmund
2007

Tag der mündlichen Prüfung:	20. November 2007
Dekan:	Professor Dr. Peter Buchholz
Gutachter:	Privatdozent Dr. Piotr Krysta
	Professor Dr. Ingo Wegener

Abstract

Combinatorial pricing encompasses a wide range of natural optimization problems that arise in the computation of revenue maximizing pricing schemes for a given set of goods, as well as the design of profit maximizing auctions in strategic settings. We consider the computational side of several different multi-product and network pricing problems and, as most of the problems in this area are NP-hard, we focus on the design of approximation algorithms and corresponding inapproximability results.

In the unit-demand multi-product pricing problem it is assumed that each consumer has different budgets for the products she is interested in and purchases a single product out of her set of alternatives. Depending on how consumers choose their products once prices are fixed we distinguish the min-buying, max-buying and rank-buying models, in which consumers select the affordable product with smallest price, highest price or highest rank according to some predefined preference list, respectively. We prove that the max-buying model allows for constant approximation guarantees and this is true even in the case of limited product supply. For the min-buying model we prove inapproximability beyond the known logarithmic guarantees under standard complexity theoretic assumptions. Surprisingly, this result even extends to the case of pricing with a price ladder constraint, i.e., a predefined relative order on the product prices. Furthermore, similar results can be shown for the uniform-budget version of the problem, which corresponds to a special case of the unit-demand envy-free pricing problem, under an assumption about the average case hardness of refuting random 3SAT-instances. Introducing the notion of stochastic selection rules we show that among a large class of selection rules based on the order of product prices the max-buying model is in fact the only one allowing for sub-logarithmic approximation guarantees.

In the single-minded pricing problem each consumer is interested in a single set of products, which she purchases if the sum of prices does not exceed her budget. It turns out that our results on envy-free unit-demand pricing can be extended to this scenario and yield inapproximability results for ratios expressed in terms of the number of distinct products, thereby complementing existing hardness results. On the algorithmic side, we present an algorithm with approximation guarantee that depends only on the maximum size of the sets and the number of requests per product. Our algorithm's ratio matches previously known results in the worst case but has significantly better provable performance guarantees on sparse problem instances. Viewing single-minded as a network pricing problem in which we assign prices to edges and consumers want to purchase paths in the network, it is proven that the problem remains APX-hard even on extremely sparse instances. For the special case of pricing on a line with paths that are nested, we design an FPTAS and prove NP-hardness.

In a Stackelberg network pricing game a so-called leader sets the prices on a subset of the edges of a network, the remaining edges have associated fixed costs. Once prices are fixed, one or more followers purchase min-cost subnetworks according to their requirements and pay the leader for all pricable edges contained in their networks. We extend the analysis of the known single-price algorithm, which assigns the same price to all pricable edges, from cases in which the feasible subnetworks of a follower form the basis of a matroid to the general case, thus, obtaining logarithmic approximation guarantees for general Stackelberg games. We then consider a special 2-player game in which the follower buys a min-cost vertex cover in a bipartite graph and the leader sets prices on a subset of the vertices. We prove that this problem is polynomial time solvable in some cases and allows for constant approximation guarantees in general. Finally, we point out that results on unit-demand and single-minded pricing yield several strong inapproximability results for Stackelberg pricing games with multiple followers.

Acknowledgements

Many people have contributed in various ways to this thesis. First and foremost, I am deeply indebted to Piotr Krysta for his consistent guidance in my research, his steady support and encouragement as an always thoughtful advisor and a friend. To Ingo Wegener I want to express my gratefulness not only for agreeing to serve as a reviewer for my thesis, but for being a great lecturer and sparking my interest in the theoretical side of computer science in the first place.

Science is not a one man show. I want to thank all my co-authors whom I was lucky enough to be able to collaborate with in the course of my studies, especially Piotr Krysta and Martin Hoefer. I am also greatly indebted to my colleagues both in Dortmund and Liverpool for stimulating a fantastic research environment and making the last three years a most enjoyable and memorable time.

This thesis would not exist without the financial support of the DFG through grant Kr 2332/1 within the Emmy Noether program.

Last, but not least, I want to thank my parents for their never faltering support throughout the last 28 years, without which none of this would have been possible.

Contents

1 Introduction

For the last decade we have been witnessing the enormous growth of the Internet as a global communication platform and its emergence as the world's number one market place. Today the Internet acts as a negotiation platform for various complex business transactions, which can be performed at a high speed and in a highly automated manner in this environment.

Among the most famous and widely cited examples of how the Internet changes business processes, we find, e.g., Google's AdWords program. In this program, the space in the 'sponsored links' section of the result page of a web search is sold to companies who wish to have their advertisement placed on the page whenever a user's search includes certain keywords. The decision which advertisements are placed on the page and in which order they will appear is essentially made by running an automated auction mechanism on companies' bids whenever the server generates a result page. However, the auction's outcome does not only depend on the amount of money each advertiser is willing to spend for a single placement, but also on a number of other factors, like maximum per day budgets, the actual combinations of keywords that trigger a certain ad, recorded click-through-rates for specific keywords and many more. As a consequence, the design of bids for the AdWords program that result in a maximum number of placements at preferably small cost has become a quite non-trivial task for ad campaigners.

The central problem, however, that arises in this context is the fact that Google and its advertisers do not at all pursue the same objective. While Google is interested in placing the most relevant ads on each page in order to keep their website as attractive as possible to potential customers, advertisers seek to optimize the value for money they receive by snatching as many placements as possible. This means that it might well be profitable for advertisers to submit bids that do not actually reflect their true preferences to the system in order to manipulate the outcome in their favor.

From the computer scientist's point of view, running the above auction is a mere optimization problem that he - difficult as it may be - is surely able to cope with. However, taking into account the strategic behavior of the participants, we look at a completely different problem. It is one of the Internet's essential characteristics that we encounter many of the computational problems we know in a less familiar game-theoretic terrain. In recent years there has been a vast amount of research aimed both at a better understanding of how the Internet operates in the absence of any central control and the question how strategic behavior can be controlled. The classic concept of stability in game theory is that of a *Nash equilibrium* [Owe95], and indeed it has been very successfully applied by computer scientists to model how networks can be built and maintained by selfishly acting users without any sort of control mechanism. In order to control strategic behavior, e.g., in an auction mechanism, Nash equilibria are somewhat problematic, since pure equilibria are not guaranteed to exist in general and, even if they do, might be rather difficult to find [FPT04]. The field of *algorithmic mechanism design* [NR99] therefore resorts to the stronger notion of *dominant strategy equilibria*, which are guaranteed to exist if one allows for some side payments, and ensure that it is always in all participants' best interest to reveal their true preferences to the mechanism. Especially the design of auction mechanisms has received considerable attention lately, as combinatorial auctions (in which several goods are auctioned simultaneously) [CSS06] essentially capture all the difficulties faced in algorithmic mechanism design in general and have therefore become the area's favorite testbed.

Aside from any game-theoretic considerations, the growth of the Internet and its influence on various aspects of business life have also fueled one of the core areas of computer science - algorithms and complexity. The Internet offers both customers and companies possibilities far beyond the scope of traditional markets and, as it becomes ever simpler to gather huge amounts of data about the market as a whole, many problems that have been on the agenda of economists for years now need to be solved efficiently and in large scale by computers.

Websites like the Product Advisor [Pro], a successor of General Motor's Auto Choice Advisor web page, allow customers to compare products and prices across the market in order to make optimal buying decisions. By asking customers about their rating of different products and their budgets, the website generates large amounts of data about consumer preferences. These data sets are extremely valuable to companies, as they allow them to streamline their product ranges and apply intelligent pricing schemes tailored to a specific market.

A problem that appears both in this context and the field of algorithmic mechanism design is that of *combinatorial multi-product pricing*. From the point of view of optimization, this problem is quite natural. Given data about a company's potential customers, revealing which product they would buy given any pricing scheme, how should the company set their prices to achieve best possible revenue? The resulting optimization problems range from very simple versions, in which each consumer is only interested in a single product, to highly involved ones, in which different products influence each other in quite unforeseen ways. Thus, multi-product pricing is an area of optimization that has to offer many challenging problems exhibiting rich combinatorial structures.

The role of multi-product pricing in the context of algorithmic mechanism design is maybe not so obvious. As mentioned before, a lot of research has been focused on the design of mechanisms for combinatorial auctions. A drawback of the classic approach of designing truthful mechanisms by applying *VCG payments* [CSS06] is the fact, that the implementation in dominant strategies comes at the price of potentially high side payments. Thus, although the mechanism is guaranteed to produce solutions that are good with respect to some global social objective, it does not necessarily generate a lot of revenue for the auctioneer. One way to circumvent this problem, at least in the case of unlimited product supply, is to resort to so-called *random sampling auctions* [BBHM05] as an alternative to VCG mechanisms. In this type of auctions, the participating bidders are randomly partitioned into two equally sized sets. Then revenue maximizing prices are computed on both sets and the products are offered at these prices to the opposite set of bidders. In this mechanism, strategic behavior trivially cannot pay off, because a bidder has no influence on the prices offered to herself, and for a sufficiently large set of bidders the revenue is guaranteed to be close to the revenue obtainable by selling the products at their optimal fixed prices.

We will focus here on the algorithmic side of multi-product pricing and investigate computational aspects of some of its combinatorially challenging versions.

1.1 Pricing for Unit-Demand Consumers

Aiming at the objective of using data as acquired by the Product Advisor website to compute intelligent pricing schemes for a company's product range, Rusmevichientong [Rus03] and Glynn et al. [GRR06] defined the *non-parametric multi-product pricing problem.* Consumers are characterized by their budgets for different products and a selection rule describing how a consumer selects a product among those she can afford once prices are fixed. Since products in this model constitute *pure substitutes* and consumers will buy exactly one product if they can afford it, they are usually referred to as *unit-demand*. Glynn et al. propose three different selection rules. In the *rank-buying* model each consumer has a ranking of all the products she is interested in. When prices are fixed she buys the highest ranked product with a price below her respective budget. In the *min-buying* and *max-buying* models a consumer buys the product with lowest or highest price not exceeding her budget, respectively. The objective of the problem is to compute revenue maximizing prices given a set of consumer samples of the above type, i.e., maximize the sum of prices of products sold to consumers from the sample under any possible price assignment. In the standard setting it is assumed that all products are available in unlimited supply.

Maybe a short discussion of our optimization goal is in order. One could argue that a more natural objective than revenue maximization might be to optimize *profit*, i.e., the sellers surplus after production costs have been deducted. However, in the case of unit-demand consumers, on which most of this work will be focused, these objectives are essentially equivalent, since we can always adjust all budgets by subtracting the production cost of the respective product and use our algorithms for revenue maximization to compute the optimal *profit margin* for each product. Unfortunately, this kind of equivalence does not hold for other versions of combinatorial pricing, in which products are not pure substitutes. In these cases, it can sometimes be rewarding to price individual products below their actual production costs in order to subsidize sales of other profitable products. In our model of computing optimal profit margins this corresponds to a problem that allows us to assign negative prices, as well, and appears to be significantly more difficult to handle than revenue maximization. For some initial results on this kind of problem the reader is referred to [BBCH07].

Rusmevichientong [Rus03] shows that the min-buying model, where each consumer has the same budget for all products she desires, allows a polynomial time algorithm, assuming a *price-ladder constraint*, i.e., a predefined total order on the prices of all products. Such a constraint is sometimes implied by the set of products in question. Aggarwal et al. [AFMZ04] present first algorithms with provable approximation guarantees for all three models: a PTAS for both rank-buying and max-buying with price-ladder, a 1.59-approximation for max-buying without price-ladder, and a logarithmic approximation in the number of consumer samples for any of the above models, assuming unlimited supply of the products.

In the more general limited supply case, in which in addition to computing product prices it has to be decided how to allocate products among consumers, a 4-approximation is derived for max-buying with price-ladder. There are many practical situations in which it is desirable to be able to handle limited supply, as well. Besides the obvious point that it might not be possible to increase production capacity beyond a certain limit, even artificially limiting product supply can sometimes be rewarding. Further results about unit-demand pricing with limited supply are presented by Guruswami et al. [GHK$^+$05], who investigate another selection rule, which is inspired by the notion of truthfulness in auction design and is first mentioned in [AFMZ04]. In the *envy-free pricing problem* a consumer buys the product that maximizes her personal utility, i.e., the difference between the product's price and her respective budget. A set of prices together with a corresponding allocation of the products is envy-free, if every consumer indeed receives the product maximizing her utility. Guruswami et al. present an algorithm with logarithmic approximation ratio for this problem, which essentially combines the result from [AFMZ04] with some interesting results from economics regarding Walrasian equilibria [KB57].

1.1.1 New Results

So far, there have been large gaps between the lower and upper bounds on the approximability of almost all versions of unit-demand pricing, the only exceptions being the max-buying model without price-ladder, for which a constant factor approximation and APX-hardness are known [AFMZ04].

As the main contribution of this work we resolve the question of approximability of most of the above unit-demand pricing models, putting emphasis on *hardness of approximation*. In particular we prove near-tight hardness results for the min-buying and max-buying models, the unit-demand envy-free pricing problem, and several versions of the rank-buying model. Many of our hardness results show the first non-constant, logarithmic, and even polynomial inapproximability for those problems. We also give algorithmic results, which close the gap in approximability of some of those models. Finally, the max-buying model is also investigated from a game theoretic perspective. Namely, we study the multi-player game obtained by assuming that the price of every product is determined by a distinct agent trying to maximize her personal revenue, and present a bound on the *price of anarchy* (cf. [KP99, Pap01]) in this game.

We will adopt the following unified notation for all considered problem variations. The unit-demand pricing problem for a set $\mathcal{C}$ of consumer samples assuming selection rule s is denoted as UDP($\mathcal{C}$)-s. Given a price-ladder constraint, we refer to the corresponding problem as UDP($\mathcal{C}$)-s-PL.

Min-Buying: We first focus on the min-buying model (UDP($\mathcal{C}$)-MIN) with unlimited supply. The best known algorithm for this problem, which simply computes the optimum solution assigning the same price to every product and is therefore termed *single-price algorithm*, has an approximation factor of H_m [AFMZ04], where m denotes the number of consumer samples. Surprisingly, it turns out that this simple algorithm is essentially best possible, as we prove that there is no $\mathcal{O}(\log^{\varepsilon} m)$-approximation algorithm for some absolute $\varepsilon > 0$, assuming NP $\not\subseteq$ DTIME($n^{\mathcal{O}(\log\log n)}$). In fact, an approximability threshold of Δ^{ε} for the independent set problem in graphs of degree at most Δ yields the same constant ε in our reduction. As so far no algorithms with approximation guarantee essentially below Δ are known for the independent set problem, this suggests that UDP($\mathcal{C}$)-MIN does not allow approximation ratios essentially better than $\log m$. We emphasize that this inapproximability result holds even in the presence of a price-ladder constraint (UDP($\mathcal{C}$)-MIN-PL). This stands in sharp contrast with the restricted version of the min-buying model in which we assume that each consumer has the same budget for all the goods she desires [Rus03], where a polynomial time algorithm follows basically by observing that in the presence of a price-ladder each consumer who is able to buy any product buys the product with smallest price according to the price-ladder. This reduces the number of products to be considered for each consumer to one and a simple dynamic programming algorithm yields the result. Remarkably, after very few natural maximization problems with logarithmic approximation threshold have been known for quite some time (see [FHKS02] for one of the first examples), UDP($\mathcal{C}$)-MIN is already the second problem from the field of product pricing (see Chapter 6 and [DFHS06] for another example) for which such a threshold can be shown.

Applying a number of small modifications our reduction also yields almost tight hardness results when the approximation ratio is expressed in terms of ℓ, i.e., the maximum number of positive budgets of any consumer, and n, the number of products. We prove that for every $\ell \geq 3$ UDP($\mathcal{C}$)-MIN and UDP($\mathcal{C}$)-MIN-PL with at most ℓ non-zero budgets per consumer are NP-hard to approximate within ℓ^{ε} for some $\varepsilon > 0$. Furthermore, in general both problems are hard to approximate within $\mathcal{O}(n^{\varepsilon})$ for some $\varepsilon > 0$, unless NP $\subseteq$ DTIME($2^{\mathcal{O}(n^{\delta})}$) for all $\delta > 0$. In addition to these lower bounds we derive some new matching algorithmic results. Specifically, there is a trivial $\mathcal{O}(n)$-approximation and an approach of Balcan and Blum

[BB06] implies an $\mathcal{O}(\ell)$-approximation for UDP($\mathcal{C}$)-MIN without price-ladder. Our hardness results are based on the classical method of graph products [BS92] to amplify the inapproximability threshold of the maximum independent set problem in bounded degree graphs. We first slightly extend the derandomized version of that construction due to Alon et al. [AFWZ95] and parametrize the maximum degree of the constructed graph product in the number of its vertices. We then encode independence in such graphs by classes of geometrically increasing budgets in our pricing problem, where vertices correspond to products. The difficulty here is that independence needs to be enforced in a somewhat asymmetric way. More precisely, based on a vertex coloring of the given graph, we can define collections of consumers that encode independence of a vertex from adjacent vertices with colors of smaller index, but we cannot do this in the opposite direction. The results concerning the min-buying model are found in Chapter 2.

Max-Buying: Among the different selection rules considered in unit-demand pricing, the max-buying model sets a counterpoint to the min-buying model and, at first glance, it might not appear as practically relevant as some of the other models. The motivation to study this problem variation stems from its connection to the economically well motivated rank-buying model, and from the fact that it turns out to be significantly more tractable than its relatives. Thus, it constitutes a promising testbed for the application of algorithmic techniques to unit-demand pricing.

The best previous algorithms for the max-buying model (UDP($\mathcal{C}$)-MAX) are presented by Aggarwal et al. [AFMZ04]. For UDP($\mathcal{C}$)-MAX with unlimited product supply they derive a 1.59-approximation based on a linear programming relaxation and randomized rounding techniques and prove that the problem is NP-hard to approximate within $16/15$. For unlimited-supply UDP($\mathcal{C}$)-MAX-PL (i.e., given a price-ladder constraint) they present a PTAS based on a rather involved dynamic programming approach, which we sketch in Section 3.4.1. However, they leave open the question whether this is the best possible algorithmic result that can be obtained in the presence of a price-ladder. We answer this question in the affirmative by proving strong NP-hardness of UDP($\mathcal{C}$)-MAX-PL.

We then consider the effect of having to deal with limited product supply. The only known result for this case is a 4-approximation for limited-supply UDP($\mathcal{C}$)-MAX-PL due to [AFMZ04]. We first have a closer look at the relation between the maximum supply and the problem's complexity. We show that without a price-ladder, limited-supply UDP($\mathcal{C}$)-MAX can be solved in polynomial time for unit-supply (i.e., given a single copy of each distinct product) but becomes APX-hard already with maximum supply of only 2. On the algorithmic side, we analyze the performance of a generic local search algorithm and prove that it yields a 2-approximation for limited-supply UDP($\mathcal{C}$)-MAX. This complements our APX-hardness result for this problem, and in fact it is the first algorithm for the limited-supply case without price-ladder with provable approximation guarantee. For unlimited supply UDP($\mathcal{C}$)-MAX our ratio does not match the best known result, which gives a 1.59-approximation [AFMZ04]. However, the previous algorithm is based on a rather problem specific LP-formulation and rounding techniques. Local search, on the other hand, appears to be a quite natural approach to a wide range of pricing problems. Seen in this light, we provide first evidence that this approach might indeed be promising also for more practical problems.

Finally, we show that our analysis of the local-search algorithm can be extended to bound the price of anarchy in the related pricing game, in which each product is owned by an individual agent setting its price. Chapter 3 deals with the max-buying model.

Rank-Buying: Many of the discussed results can be transferred to the rank-buying model. All hardness results for UDP($\mathcal{C}$)-MIN with or without price-ladder constraint hold for the rank-buying model (UDP($\mathcal{C}$)-

RANK) if we allow *non-rank-consistent budgets*, i.e., if consumers are allowed to assign higher budgets to products with lower ranks. In addition, all known algorithmic results apply here, too.

If we require *rank-consistent* budgets and consider the price-ladder case, the problem reduces to UDP($\mathcal{C}$)-MAX-PL. We prove strong NP-hardness for UDP($\mathcal{C}$)-RANK-PL as in the max-buying case, complementing the existing PTAS. Section 4.3 gives an overview of all our results for the rank-buying model.

Stochastic Selection Rules: Given a price-ladder constraint, unit-demand pricing allows a PTAS if consumers buy according to the rank-buying model and have rank-consistent budgets. This is certainly one of the strongest positive results in the field, since the rank-buying model is widely considered economically realistic and the approximation guarantee is good enough to make the algorithm applicable in practice. If the price-ladder assumption is removed, however, the current state of affairs is not satisfactory at all. On one hand, the max-buying model, which has been shown to allow constant approximation ratios, does not model rational consumer behavior, as consumers rarely tend to choose the most expensive alternative available. On the other hand, the min-buying model can be considered more realistic, but turns out to be intractable beyond logarithmic approximation guarantees, which again renders the model rather unsuitable for practical purposes. Hence, it is a major open problem to come up with economically realistic versions of unit-demand pricing that allow reasonable approximation ratios in the no price-ladder scenario. One natural approach to this task is to define a new selection rule that is sort of in-between max- and min-buying, in the sense that it is close enough to min-buying to capture rational consumer behavior, but also close enough to max-buying to be computationally tractable. We prove here that this approach is likely to fail.

To capture a wide range of selection rules that are based on product prices and are situated between the max- and min-buying models, we define the notion of *order-based stochastic selection rules*, which for each consumer define a probability distribution over the set of affordable products depending only on the relative order of prices, the problem's objective becoming maximization of the expected revenue from the resulting sales. We obtain a class of selection rules that model a wide range of consumer behavior, with max- and min-buying as the extremes at both ends of the chart. We show that constant approximation ratios are possible as long as the selection rule is close to max-buying, but become impossible under some standard complexity theoretic assumptions as soon as we make an essential step towards the min-buying objective. Especially, even the case in which a consumer chooses one of her affordable products purely at random (the *random-buying model*) turns out to be no more tractable than min-buying itself. To prove inapproximability of unit-demand pricing with stochastic selection rules, we show a probabilistic reduction from UDP($\mathcal{C}$)-MIN on a restricted class of (hard) input instances. At the core of the reduction we use a probabilistic selection procedure that with good probability finds a subset of the products on which revenue is high enough even under the min-buying objective. Our results on pricing with stochastic selection rules are found in Chapter 4.

Envy-Free Pricing: The last collection of results on unit-demand pricing considers the aforementioned special case of UDP($\mathcal{C}$)-MIN in which each consumer has the same budget for all the products she is interested in (the *uniform-budget* case). This problem can be viewed also as a special case of the unit-demand envy-free pricing problem, since with uniform budgets it is always the product with lowest absolute price that maximizes a consumer's utility. In light of the fact that this problem variation is exactly solvable in polynomial time given a price-ladder constraint, one might feel tempted to hope for improved algorithmic results in the general case, as well. As we shall see, however, such results are rather unlikely.

More precisely, we prove that assuming specific hardness of refuting random 3SAT-instances or approximating the balanced bipartite independent set problem (BBIS) in constant degree graphs, even this restricted problem version does not allow approximation guarantees essentially beyond the known logarithmic ratios. The connection between BBIS and uniform-budget UDP($\mathcal{C}$)-MIN is made via so-called *maximum expanding sequences* (MES), which can be interpreted as a combinatorial formulation of the interaction between different price levels in the pricing problem and might also be of independent interest. In order to show hardness of uniform-budget UDP($\mathcal{C}$)-MIN we need hardness of very sparse MES instances, which we obtain from constant degree BBIS (which does not have known inapproximability results) by applying the technique of derandomized graph products similar to our approach for general UDP($\mathcal{C}$)-MIN. To embed this result into a somewhat wider context, we show that it can also be derived from a hypothesis about the average case complexity of refuting random 3SAT-instances, which is essentially identical to the one put forward in [Fei02] in a similar context.

Previously, uniform-budget UDP($\mathcal{C}$)-MIN and unit-demand envy-free pricing have only been known to be APX-hard [GHK$^+$05] and settling this problem's approximation complexity is a long standing open problem. The results on uniform-budget UDP($\mathcal{C}$)-MIN are found in Chapter 5.

In several places throughout the part of this work dealing with unit-demand pricing we will also consider a slight extension of the problem, in which consumers are not represented by samples $\mathcal{C}$, but as an explicit (finite support) distribution $\mathcal{D}$. On one hand, this natural extension will allow for stronger and tighter lower bounds, while avoiding technicalities that are caused by the sampling-based representation. On the other hand, this view on the problem is widely spread in economics and is starting to receive attention in the computer science community, too [CHK07].

The table in Fig. 1.1.1 summarizes most of our results on pricing for unit-demand consumers and how they relate to previous work.

1.2 Pricing for Single-Minded Consumers

The single-minded pricing problem, which is primarily inspired by the notion of *single-mindedness* in algorithmic mechanism design, has first been considered by Guruswami et al. [GHK$^+$05]. It varies from the unit-demand model in that products now act as *pure complements* rather than substitutes. More formally, every consumer is interested in a single set of products, which she will buy if the sum of prices of products in that set does not exceed her budget.

Single-minded pricing (SMP) in general is quite comparable to the unit-demand case as far as its approximability is concerned. Guruswami et al. [GHK$^+$05] prove that the single-price algorithm yields an approximation guarantee that is logarithmic in the number m of consumers and the number n of products. An almost matching lower bound, which shows that the problem is not approximable within $\mathcal{O}(\log^{\varepsilon} m)$ (m and n are interchangeable here) for some $\varepsilon > 0$, unless NP $\subseteq$ BPTIME($2^{\mathcal{O}(n^{\delta})}$) for all $\delta > 0$, is proven by Demaine et al. [DFHS06].

For most of our discussion of SMP we will adopt a different point of view and consider the special case of pricing the edges of a network, in which consumers seek to purchase fixed paths connecting their respective source and target nodes. We can think of single-minded pricing in graphs (G-SMP) as the problem of setting up tollbooths (and defining tolls) in a privately owned highway system, for which

Variation	Previous [Lower] Upper	New Lower {Assumption}	New Upper
UDP($\mathcal{C}$)-MIN(-PL)	$\mathcal{O}(\log m)$	$\ell^{\varepsilon}\quad\forall\,\ell\geq 3$ {P $\neq$ NP} $\Omega(n^{\varepsilon})$ {NP $\not\subseteq$ DTIME($2^{\mathcal{O}(n^{\delta})}$)} $\Omega(\log^{\varepsilon} m)$ {NP $\not\subseteq$ DTIME($n^{\mathcal{O}(\log\log n)}$)}	$\mathcal{O}(\ell)$ {no PL only} $\mathcal{O}(n)$
UDP($\mathcal{C}$)-MIN {uniform budgets}	[APX-hard]	$\ell^{\varepsilon}\quad\forall\,\ell\geq\ell_0$ {R3SAT*(poly(n))-hard} $\Omega(n^{\varepsilon})$ {R3SAT*($2^{n^{\delta}}$)-hard} $\Omega(\log^{\varepsilon} m)$ {R3SAT*(poly(n))-hard}	-
UDP($\mathcal{C}$)-MAX	[16/15], 1.59 (LP-based)	–	2 (combinatorial)
UDP($\mathcal{C}$)-MAX {limited supply}	[–], –	APX-hard {supply ≥ 2} in P {supply ≤ 1}	2
UDP($\mathcal{C}$)-MAX-PL {limited supply}	[–], 4	strongly NP-hard	–
UDP($\mathcal{C}$)-MAX-PL	[–], PTAS	strongly NP-hard	–

Figure 1.1: Our results on unit-demand pricing. Results apply to unlimited supply, unless stated otherwise. Hardness results with ε and complexity assumptions with δ are assumed to hold for some ε, $\delta > 0$. For a definition of R3SAT$^{\star}(t(n))$-hardness see Section 5.2.

reason the problem has been termed *tollbooth problem* in [GHK+05]. Guruswami et al. show that G-SMP is APX-hard in general and develop polynomial time algorithms for several special cases, e.g., if the underlying network is a rooted tree and all paths share the root as a common starting point. A special case that has received a lot of attention is that of networks consisting of a single line, the so-called *highway problem*. In [GHK+05] it is shown how to solve this problem optimally when all paths are of constant length or all budgets are of at most constant size. Improved approximation results for the highway problem are presented by Balcan and Blum [BB06] and Elbassioni et al. [ESZ07], whose quasi-PTAS is the best known algorithm for highway pricing to date.

1.2.1 New Results

We will present two types of results on SMP. Guruswami et al. [GHK+05] prove that G-SMP is APX-hard. However, their reduction creates a problem instance in which some of the products are requested by a constant fraction of all consumers. From a technical standpoint, this appears quite unavoidable, since an approximation preserving reduction to G-SMP always brings up the problem that we need to force optimal (or approximately optimal) solutions to be in a sense integral in order to be able to reconstruct solutions to the combinatorial problem that is our reduction's starting point. On the other hand, it is certainly desirable to have hardness results also for sparse instances, especially because it turns out that the number of requests per product and the maximum number of products requested by any consumer are the most crucial parameters when it comes to finding good approximations using upper bounding techniques known so far.

Our first main result is a proof of APX-hardness even if the parameters mentioned above (and several more) are bounded by a small constant. The result is based on a reduction from MAX-SAT with clauses of length 2 and essentially requires the design of gadgets for the pricing instance that simulate the behavior of clauses in the MAX-SAT instance. Along the way we design smaller gadgets that not only model the behavior of literals, but also enable us to prove a first lower bound for the highway pricing problem, which turns out to be NP-hard. Interestingly, this holds even on instances in which all requested paths are *nested*, i.e., either disjoint or completely contained inside each other. For this restricted version we show how to obtain an FPTAS based on a dynamic programming approach.

Looking at general SMP we argue that our results on unit-demand envy-free pricing can be extended to the single-minded case. Specifically, we complement known hardness results by proving that SMP is not approximable within $\mathcal{O}(n^{\varepsilon})$ for some ε under our assumption about average case hardness of random 3SAT.

Our second main result is an approximation algorithm for SMP that asymptotically matches the approximation guarantee of the single-price algorithm in the worst case, but is capable of exploiting the special structure of sparse problem instances to outperform the single-price approach in these cases. Guruswami et al. [GHK+05] prove that the single-price algorithm has approximation ratio $H_m + H_n$. The approximation guarantee of our algorithm, which is purely combinatorial and based on a partitioning approach, is $\mathcal{O}(\log \ell + \log B)$, where ℓ is an upper bound on the number of products requested by any consumer and B denotes the maximum number of consumers interested in any single product. Note, that clearly $\ell \leq n$ and $B \leq m$ on any problem instance. Chapter 6 considers pricing for single-minded consumers.

1.3 Stackelberg Pricing

In the last chapter we turn to the most general form of network and multi-product pricing. Chapters 2 through 6 consider pricing for quite restrictive types of consumers, who are interested in purchasing exactly one product in the pure substitutes or unit-demand case and exactly one subset of products in the pure complements or single-minded scenario. In general, however, it is easy to imagine that consumers act according to much more complex preferences and even modelling general consumer behavior seems quite unachievable. We adopt here the notion of a *general bidder* from algorithmic mechanism design that is defined along the lines of utility maximization and is a natural generalization of the unit-demand envy-free pricing problem. A consumer is characterized by a *valuation function* that assigns a valuation to every possible subset of products. Once prices are fixed, each consumer buys the subset of products that maximizes her utility, i.e., the difference between her valuation for that subset and the sum of prices of products in the set.

Once again we view the problem as the task of pricing the links (or nodes) in some underlying network, in which consumers seek to purchase subnetworks according to their requirements. In our setting, a *leader* is allowed to assign prices only to a subset of the network links (the *pricable* edges), the prices of the remaining (*fixed-price*) edges are fixed. Once prices have been assigned to all edges by the leader, each consumer (termed *follower*) purchases the cheapest subnetwork satisfying her requirements and the leader receives payments according to the prices of pricable edges bought by the followers. The subnetworks required by the followers can be paths connecting specific vertices, spanning trees or anything else. The only assumption that we will make here is that each follower is able to find her cheapest feasible subnetwork in polynomial time.

This type of pricing problem, in which preferences are implicitly defined in terms of some optimization problem, is usually referred to as *Stackelberg pricing* [vS34]. In the standard 2-player form we are given a leader setting the prices on a subset of the network and a single follower seeking to purchase some kind of min-cost subnetwork.

It should be pointed out that Stackelberg pricing with multiple followers captures multi-product pricing with arbitrary valuation functions in full generality, if we drop the assumption that followers are limited to polynomial time computations but allow general bidders represented by oracles. This is true because we can encode general valuation functions by using pricable edges as products and fixed-price edges in combination with the feasibility constraint to encode budgets. We note that all our results on general Stackelberg games also hold for multi-product pricing among consumers with general valuation functions.

1.3.1 New Results

The single-follower Stackelberg network pricing problem has first been considered for a follower seeking to purchase a path connecting a fixed pair of vertices [LMS98]. Roch et al. [RSM05] present an approximation algorithm for this problem, which achieves approximation guarantee $\mathcal{O}(\log m)$, where m is the number of pricable edges, and relies on a quite problem specific recursive branching approach. Quite recently, Cardinal et al. [CDF$^+$07] investigated the corresponding minimum spanning tree game, in which the follower buys a min-cost spanning tree when prices are fixed, obtaining a logarithmic approximation guarantee by applying the single-price algorithm. Moreover, they prove that this algorithm is even more widely applicable and yields similar approximation guarantees for all matroid based Stackelberg games, i.e., pricing games in which the feasible subnetworks of the follower form the basis of a matroid.

We present a generalization of this result to general Stackelberg games. The previous limitation to matroids stems from the difficulty to determine the necessarily polynomial number of candidate prices that can be tested by the algorithm. We develop a novel characterization of the small set of *threshold prices* that need to be tested and obtain a polynomial time $(1+\varepsilon)H_m$-approximation for arbitrary $\varepsilon > 0$. Our analysis turns out to be perfectly tight for shortest path as well as minimum spanning tree games. This result generalizes to the case of multiple followers, in which the approximation ratio becomes $(1+\varepsilon)(H_m + H_k)$, where k denotes the number of followers. This can be shown to be essentially best possible by an approximation preserving reduction from single-minded pricing.

An even more general version of the problem, which has been considered before by Roch et al. [RSM05] and Bouhtou et al. [BGvH$^+$04], is the case of multiple *weighted* followers, which arises naturally in network settings where different followers come with different routing demands. Here the profit from selling an edge is defined as the edge's price multiplied with the demand a follower routes along it. It has been conjectured before that no approximation essentially better than the number of followers is possible in this scenario. We disprove this conjecture by presenting an alternative analysis of the single-price algorithm resulting in an approximation ratio of $(1 + \varepsilon)m^2$. Additionally, we derive a lower bound of $\mathcal{O}(m^\varepsilon)$ for the weighted player case, which yields instances of the so-called *river tarification problem* and resolves a previously open problem from [BGvH$^+$04].

It is a central open question how the type of subnetwork purchased by the follower influences the complexity of the resulting 2-player Stackelberg game. We present a first example in which better than logarithmic approximation guarantees are achievable by using algorithmic techniques similar to those that yield polynomial time algorithms for the optimization problem solved by the follower. In the 2-player bipartite vertex cover game, the leader sets prices on a subset of the vertices of a bipartite graph, the follower purchases a min-cost vertex cover of the graph's edges. As it turns out, max-flow related techniques yield a polynomial time algorithm for revenue maximization in this game, if all pricable vertices are on the same side of the bipartition. We first derive an upper bound on the possible revenue in terms of the min-cost vertex cover not using any pricable vertices and the minimum portion of fixed cost in any possible cover. Using iterated max-flow computations, we then determine a pricing with total revenue that eventually coincides with our upper bound. This algorithm easily extends to a 2-approximation for the general 2-player bipartite vertex cover game. Our results on Stackelberg pricing are found in Chapter 7.

1.4 List of Publications

This thesis is based on the following publications:

- Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing.
  Joint work with Piotr Krysta.
  In *Proc. of 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

  Results published in this paper are found in Chapters 2, 3 and the last part of Chapter 4.

- Towards Hardness of Envy-Free Pricing.
  ECCC Technical Report TR06-150, 2006.

  Results published in this paper are found in Chapter 5.

- Single-Minded Unlimited-Supply Pricing on Sparse Instances.
  Joint work with Piotr Krysta.
  In *Proc. of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

  Results published in this paper are found in Chapter 6.

- Stackelberg Network Pricing Games.
  Joint work with Martin Hoefer and Piotr Krysta.
  Technical Report ULCS-07-022, The University of Liverpool, 2007.

  Results published in this paper are found in Chapter 7.

2 Buying Cheap is Expensive: The Min-Buying Model

We start our investigation of algorithmic pricing by looking at the unit-demand case, in which products are pure substitutes and every consumer seeks to purchase a single product out of a set of alternatives subject to individual budget constraints. This type of consumer behavior, which was first considered in [Rus03], constitutes a whole class of pricing problems varying in the way consumers select the product they are going to buy once prices are fixed.

A natural way to define the selection process is to assume that each consumer simply selects the cheapest product that does not violate her budget constraint. This version of the problem, which is usually referred to as unit-demand *min-buying*, is quite attractive as it essentially corresponds to our intuitive understanding of rational consumer behavior. However, as we shall see this comes at the price of high approximability thresholds that pose an insurmountable barrier if one insists on practically relevant algorithmic results in the general case.

It is therefore necessary to search for restricted problem versions if we want to stand a chance of achieving near optimal pricing schemes. One such restriction, which has been considered before is the so-called *price-ladder constraint*, which imposes a relative order on the prices of all products. While this approach has been proven to drastically reduce the complexity of some other versions of unit-demand pricing, it turns out that it is not very successful in the min-buying scenario. Another standard approach to simplify the problem is to assume that each consumer is only interested in a small number of the available products and, as we shall see, this is in fact a possibility to achieve improved approximation guarantees in some cases.

This chapter is organized as follows. In Section 2.1 we give a formal definition of unit-demand pricing in its most general form, which is then refined to the min-buying model. Section 2.2 introduces one of the folklore results in algorithmic pricing due to [AFMZ04], the very simple single-price algorithm, in the context of general unit-demand pricing. The main results of this chapter are found in Section 2.3, which shows that despite its simplicity the single-price algorithm is in fact essentially best possible for the min-buying model. Interestingly, the presented reductions yield (near) tight inapproximability results for quite different parametrizations of the approximation guarantee and hold even under the presence of a price-ladder constraint. To show tightness of our bounds in the case of sparse problem instances, Section 2.4 sketches how a previously known algorithm can be used to obtain tight approximation ratios expressed in the number of non-zero budgets per customer. Section 2.5 gives an overview of related literature.

2.1 Preliminaries

In the computer scientist's *Unit-Demand Pricing Problem* with unlimited product supply we are given a universe of products $\mathcal{U}$, $|\mathcal{U}| = n$, and consumer samples $\mathcal{C}$, $|\mathcal{C}| = m$, consisting of budgets $b(c, u) \in \mathbb{R}_0^+$, where $b(c, u)$ denotes the maximum amount consumer $c \in \mathcal{C}$ is willing to spend on product u. For a price assignment $p : \mathcal{U} \to \mathbb{R}_0^+$ we let

$$A_c(p) = \{u \in \mathcal{U} \mid p(u) \leq b(c,u)\}$$

refer to the set of products consumer c can afford to buy and

$$\mathcal{A}(p) = \{c \in \mathcal{C} \mid A_c(p) \neq \emptyset\}$$

denote the set of consumers that can afford to buy any product under p. Given fixed prices p, a *selection rule*

$$s : \mathcal{C} \times (\mathcal{U} \to \mathbb{R}_0^+) \to \mathcal{U} \cup \{\varnothing\}$$

assigns to each consumer c the product $s(c,p) \in A_c(p) \cup \{\varnothing\}$ she is going to buy under p. If $s(c,p) = \varnothing$, then c does not buy any product. For ease of notation we define $p(\varnothing) = 0$.

In the computer scientist's or *sampling-based* unit-demand pricing problem with selection rule s (denoted as UDP($\mathcal{C}$)-s) we want to find prices p that maximize the revenue

$$r_s(p) = \sum_{c \in \mathcal{C}} p\big(s(c,p)\big).$$

A natural extension of the problem is obtained if we assume that our knowledge of consumer preferences is not obtained from some sampling procedure, but that we know the explicit probability distribution over the space $\mathcal{C}^*$ of all possible consumers, which is a widely spread assumption in economics. Thus, in the economist's or *distribution-based* unit-demand pricing problem UDP($\mathcal{D}$)-s we are given probability distribution $\mathcal{D}$ over consumer space $\mathcal{C}^*$. We will mostly focus on lower bounds for this model, in which case we may w.l.o.g. restrict ourselves to finite support distributions, i.e., we are given a set of consumers $\mathcal{C}$ and a discrete distribution $\mathcal{D}$ on $\mathcal{C}$. Our goal is to find prices p maximizing the expected revenue

$$r_s(p) = \sum_{c \in \mathcal{C}} \Pr_{\mathcal{D}}\big(c\big) \cdot p\big(s(c,p)\big).$$

from a sale to a single consumer drawn according to distribution $\mathcal{D}$.

Choosing as selection rule

$$s_{\min}(c,p) = \begin{cases} \operatorname{argmin}\{p(u) \mid u \in A_c(p)\}, & \text{if } A_c(p) \neq \emptyset \\ \varnothing, & \text{else} \end{cases}$$

we obtain the min-buying model, in which each customer chooses to purchase the least expensive product with a price not exceeding her respective budget. We next give a compact formal definition of the sampling-based unit-demand min-buying problem UDP($\mathcal{C}$)-MIN. The definition of the distribution-based problem is analogous.

Definition 2.1.1. *Given products $\mathcal{U}$, $|\mathcal{U}| = n$, and consumer samples $\mathcal{C}$, $|\mathcal{C}| = m$, consisting of budgets $b(c,u) \in \mathbb{R}_0^+$,* UDP($\mathcal{C}$)-MIN *asks to find a price assignment $p : \mathcal{U} \to \mathbb{R}_0^+$ maximizing*

$$r_{\min}(p) = \sum_{c \in \mathcal{A}(p)} \min\{p(u) \mid u \in A_c(p)\},$$

with $A_c(p)$ and $\mathcal{A}(p)$ as defined above.

Previous literature sometimes considers unit-demand pricing under the restriction of a so-called *price-ladder constraint*, i.e., a predefined order of the product prices. Such constraints naturally arise in scenarios where products are directly comparable and pricing a product above some superior alternative cannot be profitable. As an example, imagine a company that wants to price their range of MP3-players, which vary only in hard disc space.

Definition 2.1.2. *Given products $\mathcal{U} = \{u_1, \dots, u_n\}$ a price-ladder constraint (PL) $\pi \in S_n$ is a relative order $p(u_{\pi(1)}) \leq \cdots \leq p(u_{\pi(n)})$ on the product prices.* UDP($\mathcal{C}$)-s-PL *asks for a revenue maximizing price assignment p satisfying this constraint.*

2.2 The Single-Price Algorithm

Given a collection of consumer samples it is not difficult to argue that the prices assigned to the products in an optimal solution are always chosen from the set of distinct budget values. The *single-price algorithm* (Algorithm 1), which was originally proposed by Aggarwal et al. [AFMZ04], works by trying the maximum budget of each consumer as a price for all the products and returning the best such uniform price assignment. Theorems 2.2.1 and 2.2.2 analyze the approximation guarantee of Algorithm 1 expressed in terms of both the number of consumer samples and products. The proof of Theorem 2.2.1 is included for the sake of completeness.

Algorithm 1: The single-price algorithm for unit-demand consumers.

For each consumer $c \in \mathcal{C}$ let $b_c = \max_u b(c, u)$.
Define price assignment $p_c : \mathcal{U} \to \mathbb{R}_0^+$ by $p_c(u) = b_c$ for all $u \in \mathcal{U}$.
Return the best such price assignment $p = \text{argmax}\{r_{\min}(p_c) \,|\, p_c : c \in \mathcal{C}\}$.

Theorem 2.2.1 ([AFMZ04])**.** *The single-price algorithm computes an H_m-approximation with respect to optimal revenue for* UDP($\mathcal{C}$)-s *and* UDP($\mathcal{C}$)-s-PL. *This bound is tight.*

Proof: Let consumer samples $\mathcal{C} = \{c_1, \dots, c_m\}$ be given and assume w.l.o.g. that $b_{c_1} \geq \cdots \geq b_{c_m}$, where $b_{c_j} = \max_u b(c_j, u)$ as in the algorithm. Furthermore, let $r_{\min}(p^*)$ denote the revenue obtained by the optimal price assignment on this problem instance. It clearly holds that $r_{\min}(p^*) \leq \sum_{j=1}^{m} b_{c_j}$, since no consumer can buy a product at a price above her maximum budget value. Let then p be the price assignment returned by the single-price algorithm and observe that $r_{\min}(p) \geq j \cdot b_{c_j}$ for $j = 1, \dots, m$, since at price b_{c_j} consumers $c_1, \dots, c_j$ can afford to buy some product. We conclude that

$$\begin{aligned} H_m \cdot r_{\min}(p) &= \sum_{j=1}^{m} \frac{r_{\min}(p)}{j} \geq \sum_{j=1}^{m} \frac{j \cdot b_{c_j}}{j} \\ &= \sum_{j=1}^{m} b_{c_j} \geq r_{\min}(p^*), \end{aligned}$$

which proves the upper bound. Clearly, a price assignment returned by the single-price algorithm trivially satisfies any price-ladder constraint. To see that the bound is in fact tight, consider an instance with m

consumers $c_1, \dots, c_m$, products $u_1, \dots, u_m$ and budgets $b(c_j, u_j) = 1/j$, $b(c_j, u_k) = 0$ for all $j \neq k$. The optimal price assignment $p^*(u_j) = 1/j$ yields revenue $\sum_{j=1}^{m} 1/j = H_m$, whereas every price assignment consisting of a single price results in total revenue at most 1. □

Theorem 2.2.2. *The single-price algorithm computes an n-approximation with respect to optimal revenue for* UDP($\mathcal{C}$)-s *and* UDP($\mathcal{C}$)-s-PL*. This bound is tight.*

Proof: Given an optimal price assignment p^*, let $C^*(u)$ denote the set of consumers buying product $u \in \mathcal{U}$. Thus, the revenue made by selling u in the optimal solution can be written as $|C^*(u)| \cdot p^*(u)$. Now observe that clearly $b_c \geq p^*(u)$ for every $c \in C^*(u)$, where $b_c = \max_u b(c, u)$, and, for $c' = \operatorname{argmin}\{b_c \mid c \in C^*(u)\}$, we get that $r_{\min}(p_{c'}) \geq |C^*(u)| \cdot b_{c'} \geq |C^*(u)| \cdot p^*(u)$. Summing over all products u yields the upper bound.

For tightness consider an instance with n products $u_1, \dots, u_n$. Let $k \in \mathbb{N}$. For each product u_j we define a collection of k^j customers, each with budget k^{-j} for u_j and budget 0 for all other products. It is straightforward to argue that the optimal price assignment $p^*(u_j) = k^{-j}$ results in overall revenue n, while the single-price algorithm returns price assignment $p(u_j) = k^{-n}$ for all j with revenue

$$r_{\min}(p) = \sum_{j=0}^{n-1} k^{-j} = \frac{1 - k^{-n}}{1 - k^{-1}} \to 1$$

for $k \to +\infty$, which finishes the proof. □

2.3 Hardness of Approximation

We now leave general unit-demand pricing behind and focus on the min-buying model. Having seen that as simple an approach as the single-price algorithm is already enough to obtain provable approximation guarantees, it is a most natural question to ask whether more elaborate algorithmic techniques can be used to achieve better results for UDP($\mathcal{C}$)-MIN. Surprisingly, it turns out that this is not the case and the approximation results presented in Section 2.2 are essentially the best one can hope for.

We will prove hardness results based on an approximation preserving reduction of the *Independent Set Problem* (IS).

Definition 2.3.1. *Given an undirected graph $G = (V, E)$, the* Independent Set Problem *(*IS*) asks for a maximum cardinality subset of pairwise non-adjacent vertices, i.e., a maximum cardinality subset $V' \subseteq V$ with $\{v, w\} \notin E$ for all $v, w \in V'$. We denote the size of a maximum independent set in G by $\alpha(G)$.*

As one of the most fundamental problems in combinatorics, several tight algorithmic and inapproximability results are known for both general IS and restricted problem versions, e.g., in bounded degree graphs. The best known lower bounds include inapproximability within $\mathcal{O}(n^{1-\varepsilon})$, unless P = NP, in general graphs on n vertices [Zuc06] and inapproximability within $\Delta / \log^{\mathcal{O}(1)} \Delta$ in graphs of maximum degree Δ [ST06] assuming that the *unique games conjecture* [Kho02] is true.

The high-level idea of the reduction is as follows. For a graph with vertices $v_1, \ldots, v_n$ we define a corresponding universe of products $u_1, \ldots, u_n$ and associate with each product u_j a threshold price p_j, where the p_j's essentially form a geometrically increasing sequence. Then, for each product u_j, we introduce a collection of p_j^{-1} consumers with budgets equal to the threshold prices for u_j itself and products with smaller indices corresponding to adjacent vertices. Budgets for all other products are 0. These collections of consumers encode independence in the sense that consumers associated with product j can generate reasonable revenue only by buying product u_j (their budgets for all other products being too small) and this is only possible if prices of products corresponding to adjacent vertices are set in a way that does not allow them to generate any revenue at all.

The above reduction applied to general IS immediately yields hardness of distribution-based UDP($\mathcal{D}$)-MIN (see Theorem 2.3.11), since in this case we can express exponentially large collections of identical consumers by a single consumer with appropriate probability in the finite support distribution. This, however, is not feasible for the sampling-based problem variation, in which we must guarantee that the size of the consumer sample can be bounded in terms of the size of the graph. Here, the key to success lies in looking at bounded degree instances of IS, which allow for efficient encoding in terms of small consumer samples by a simple coloring argument. The main issue is finding the right degree bound in order to obtain tight hardness results while not letting IS instances grow too dense to be encoded in terms of UDP($\mathcal{C}$)-MIN.

2.3.1 Independent Sets and Derandomized Graph Products

The theory of inapproximability is based on the fundamental PCP-Theorem [ALM+98]. Loosely speaking, this theorem in its combinatorial form states that not only is it NP-hard to decide whether a given 3-SAT formula has a satisfying assignment, but even distinguishing the cases that it is satisfiable or every assignment satisfies at most a constant fraction of the formula's clauses is similarly difficult. Using this as a starting point one obtains NP-hardness results for so-called *gap-versions* of various other problems, among them bounded-degree IS [PY91]. Formally, let $\mathcal{G}_a$ and $\mathcal{G}_b$ be two families of graphs with maximum degree bounded by 3 and $\alpha(G) \leq an$ for $G \in \mathcal{G}_a$, $\alpha(G) \geq bn$ for $G \in \mathcal{G}_b$.

Proposition 2.3.2 ([PY91])**.** *There exist constants $0 < a < b < 1$, such that given $G \in \mathcal{G}_a \cup \mathcal{G}_b$ it is* NP-*hard to decide whether $G \in \mathcal{G}_a$ or $G \in \mathcal{G}_b$.*

IS exhibits the interesting property that it allows super-constant hardness of approximation results based on Proposition 2.3.2 basically by reducing the problem to itself in a clever way, thereby increasing the gap between yes- and no-instances. The following defines the reduction used to achieve this kind of gap amplification.

Definition 2.3.3 ([BS92])**.** *Let $G = (V, E)$ be a graph and $k \in \mathbb{N}$. We define the k-fold* graph product *$G^k = (V^k, E_k)$ of graph G by $V^k = V \times \cdots \times V$ and $\{(v_1, \ldots, v_k), (w_1, \ldots, w_k)\} \in E_k$ if and only if $\{v_1, \ldots, v_k, w_1, \ldots, w_k\}$ is not an independent set in G.*

How do graph products help to get stronger inapproximability results for IS? Let a graph $G \in \mathcal{G}_a \cup \mathcal{G}_b$ be given and consider G^k for $k = \Theta(\log n)$. If $G \in \mathcal{G}_a$, then the maximum independent set in G^k is of size $(an)^k$. If $G \in \mathcal{G}_b$, then G^k has an independent set of size $(bn)^k$. Thus, the gap has increased to $(bn)^k/(an)^k = n^\delta$, where $\delta = \log(b/a)$. Observe that G^k has n^k vertices, which is not polynomial in n

for our choice of k. In order to circumvent this problem, one does not construct G^k explicitly, but samples $(1/a)^k$ of its vertices independently at random instead. The resulting so-called *randomized graph product* [BS92] is of polynomial size in n and preserves (almost) the same gap with high probability. Thus, given a polynomial time $\mathcal{O}(n^\delta)$-approximate algorithm for IS for sufficiently small $\delta > 0$, we can distinguish the cases $G \in \mathcal{G}_a$ and $G \in \mathcal{G}_b$ in polynomial time.

Randomized graph products are a way to show inapproximability of general IS. However, the randomized nature of the construction does not allow hardness amplification while enforcing a strict degree bound. The key to bounded-degree instances lies in replacing the randomized sampling procedure and considering *derandomized graph products* [AFWZ95] instead. Given graph $G = (V, E)$, we construct a non-bipartite d-regular *Ramanujan graph* H, which has the same vertices as G and constant degree d that depends only on a and b. Vertices of the derandomized graph product DG^k are obtained by choosing a vertex of H uniformly at random and taking a random walk of length $k-1$ starting at this vertex. For $k = O(\log n)$ the number nd^{k-1} of such random walks is polynomial and, thus, DG^k can be constructed deterministically in polynomial time. The edges of DG^k are defined as before. Now let dA be the (symmetric) adjacency matrix of H, where $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{n-1}$ are eigenvalues of matrix A, and let $\lambda = \max\{\lambda_1, |\lambda_{n-1}|\}$. The following theorem gives an upper and lower bound on the size of the maximum independent set in DG^k.

Theorem 2.3.4 ([AFWZ95]). *For any graph G and any k it holds that*

$$\alpha(G)d^{k-1}\left(\frac{\alpha(G)}{n} - \lambda\right)^{k-1} \leq \alpha(DG^k) \leq \alpha(G)d^{k-1}\left(\frac{\alpha(G)}{n} + \lambda\right)^{k-1}.$$

Theorem 2.3.5 shows the application of derandomized graph products to obtain hardness of bounded-degree IS. Although essentially the theorem itself is not new, we include the proof to show that we can express the super-constant maximum degree of DG^k in terms of the number of its vertices, which will be needed for the reduction to UDP($\mathcal{C}$)-MIN.

Theorem 2.3.5. *For any non-decreasing function $f : \mathbb{N} \longrightarrow \mathbb{R}^+$ with $f(n) \leq n$ and $f(n^c) \leq f(n)^c$ for all $c \geq 1$, $n \in \mathbb{N}$, let $\mathcal{G}_f$ be the family of graphs $G = (V, E)$, $|V| = n$, with maximum degree $\Delta = \mathcal{O}(f(n))$. There exists a constant $\varepsilon > 0$, such that it is* NP*-hard to approximate $\alpha(G)$ within $\mathcal{O}(f(n)^\varepsilon)$ for $G \in \mathcal{G}_f$.*

Proof: Let $\mathcal{G}_a$ and $\mathcal{G}_b$ be defined as above and let $G \in \mathcal{G}_a \cup \mathcal{G}_b$, $G = (V, E)$, $|V| = n$. Choosing $0 < a < b < 1$ appropriately it is NP-hard to decide whether $G \in \mathcal{G}_a$ or $G \in \mathcal{G}_b$ by Proposition 2.3.2. We now consider the k-fold derandomized graph product $DG^k = (DV, DE)$.

By its construction we have that $|DV| = nd^{k-1}$. Let $(v_1, \ldots, v_k) \in DV$ and assume that there are indices i and j, such that $\{v_i, v_j\} \in E$. In this case it follows that $\{(v_1, \ldots, v_k), (w_1, \ldots, w_k)\} \in DE$ for all $(w_1, \ldots, w_k)$. Thus, DG^k contains a number of vertices of degree $nd^{k-1} - 1$. We define the modified graph $mDG^k = (mDV, mDE)$ by removing all these vertices from DG^k. We observe that $\alpha(mDG^k) = \alpha(DG^k)$. By Theorem 2.3.4 an independent set of size bn in G results in an independent set of size at least $bnd^{k-1}(b - \lambda)^{k-1}$ in DG^k. If less than this number of vertices are contained in mDG^k, it follows that $G \in \mathcal{G}_a$. Thus, w.l.o.g. we may assume that

$$bnd^{k-1}(b - \lambda)^{k-1} \leq |mDV| \leq nd^{k-1}.$$

In mDG^k an edge $\{(v_1,\dots,v_k),(w_1,\dots,w_k)\}$ exists only if there are indices i and j, such that $\{v_i,w_j\}\in E$. We fix $(v_1,\dots,v_k)$ and count the maximum number of adjacent vertices. There are k^2 possibilities to select i and j. Fixing indices fixes v_i as well and, by the fact that G has maximum degree 3, there are at most 3 possible choices for w_j. Finally, there remain d^{k-1} possibilities to choose the random walk generating $(w_1,\dots,w_k)$. Thus, mDG^k has maximum degree $\Delta \le 3k^2d^{k-1}$.

It is known that for sufficiently many n and d one can construct d-regular Ramanujan graphs with good expansion rate or, equivalently, adjacency matrices with small second largest eigenvalue.

Fact 2.3.6 ([HLW06]). *For infinitely many values of n and d one can construct a d-regular Ramanujan graph, such that $\lambda \le 2\sqrt{d-1}/d$.*

For the remainder of the proof, let us assume that d-regular Ramanujan graph H with $d \ge 36/(b-a)^2$ is chosen as described in Fact 2.3.6. Furthermore, we choose

$$k = (1-\mu)\log_d f(n)$$

for some $0<\mu<1$. We need to show that the resulting derandomized graph product on N vertices has maximum degree $\mathcal{O}(f(N))$ and the gap is amplified to $\Omega(f(N)^\varepsilon)$ for some constant $\varepsilon>0$.

Size of mDG^k: Note that by construction $\frac{3}{2}b^{-1} < d$, $\lambda \le b/3$ and $c>1$. Thus, the number of vertices N of mDG^k is lower bounded by

$$N \ge bnd^{k-1}(b-\lambda)^{k-1} \ge bn\left(\frac{d}{\frac{3}{2}b^{-1}}\right)^{k-1} = \Omega(n).$$

On the other hand, by our choice of k we obtain an upper bound of

$$N < nd^k = nf(n)^{1-\mu} = \mathcal{O}(n^2),$$

where we use that $f(n)\le n$ by assumption.

Degree of mDG^k: The maximum degree Δ of mDG^k is upper bounded by

$$\Delta \le 3k^2d^{k-1} \le 3((1-\mu)\cdot\log_d f(n))^2 f(n)^{1-\mu} = \mathcal{O}(f(N)),$$

using that $\log^2 f(n) = o(f(n)^\mu)$, $N=\Omega(n)$ and f is non-decreasing.

Gap Amplification: By the fact that $d \ge 36/(b-a)^2$ we have that

$$\lambda < \frac{2}{\sqrt{d}} \le \frac{1}{3}(b-a).$$

By Theorem 2.3.4 the gap between the cases that $G\in\mathcal{G}_a$ and $G\in\mathcal{G}_b$ is then amplified to

$$\frac{bnd^{k-1}(b-\lambda)^{k-1}}{and^{k-1}(a+\lambda)^{k-1}} \ge \left(\frac{b-\lambda}{a+\lambda}\right)^k > \left(\frac{(a+\lambda)+\lambda}{a+\lambda}\right)^k > (1+\lambda)^k.$$

Choosing a small enough constant γ, such that $(4/\lambda^2)^\gamma \le (1+\lambda)$ and using that by Fact 2.3.6 $d<4/\lambda^2$, the gap size is lower bounded by

$$(1+\lambda)^k \ge (4/\lambda^2)^{\gamma k} > d^{\gamma k}.$$

Plugging in the definition of k and using that $N = \mathcal{O}(n^2)$ and $f(n^2) \leq f(n)^2$ we get

$$d^{\gamma k} = d^{\gamma(1-\mu)\log_d f(n)} = f(n)^{\gamma(1-\mu)} = \Omega(f(N)^{\varepsilon}),$$

where $\varepsilon = \gamma(1-\mu)/2$. □

2.3.2 Reduction to UDP($\mathcal{C}$)-MIN

Theorem 2.3.7 introduces a general approximation preserving reduction from IS in graphs of bounded degree $\mathcal{O}(f(n))$ to UDP($\mathcal{C}$)-MIN with or without price-ladder constraint. Corollaries 2.3.8, 2.3.9 and 2.3.10 derive hardness of approximation results for approximation guarantees expressed in various parameters under standard complexity theoretic assumptions. Corollary 2.3.11 uses Theorem 2.3.7 to derive an even stronger inapproximability result for distribution-based UDP($\mathcal{D}$)-MIN.

Theorem 2.3.7. *Let $f : \mathbb{N} \longrightarrow \mathbb{R}^+$ be a non-decreasing function with $f(n) \leq n$ and $f(n^c) \leq f(n)^c$ for all $c \geq 1$.* UDP($\mathcal{C}$)-MIN *and* UDP($\mathcal{C}$)-MIN-PL *are not approximable in polynomial time within $\mathcal{O}(f(n)^{\varepsilon})$ for some $\varepsilon > 0$ on instances with*

- *n different products and*
- *$m = \mathcal{O}(n4^{f(n)}f(n)^{f(n)})$ customers,*
- *each having $\mathcal{O}(f(n))$ non-zero budgets,*

unless NP $\subseteq$ DTIME($nf(n)^2 4^{f(n)} f(n)^{f(n)}$).

Proof: Consider the family $\mathcal{G}_f$ of graphs $G = (V, E)$, $|V| = n$, with degree bounded by $\mathcal{O}(f(n))$. By Theorem 2.3.5 it is NP-hard to approximate $\alpha(G)$ for $G \in \mathcal{G}_f$ within $\mathcal{O}(f(n)^{\varepsilon})$ for some $\varepsilon > 0$. Towards a contradiction, we assume that there is a polynomial time algorithm with approximation guarantee $\mathcal{O}(f(n)^{\varepsilon})$ for UDP($\mathcal{C}$)-MIN-PL. For a given graph $G = (V, E)$ from $\mathcal{G}_f$ let Δ denote its maximum degree. Clearly, we can compute a $(\Delta + 1)$-coloring of the vertices of G, which we denote by $V = V_0 \cup \ldots \cup V_{\Delta}$. For ease of notation let $V_i = \{v_{i,j} \mid j = 0, \ldots, |V_i| - 1\}$. Furthermore, by

$$\mathcal{V}(v_{i,j}) = \{v_{k,\ell} \mid \{v_{i,j}, v_{k,\ell}\} \in E \text{ and } k < i\}$$

we refer to the vertices that are adjacent to $v_{i,j}$ and belong to a color class with index smaller than i. We define a corresponding instance of UDP($\mathcal{C}$)-MIN-PL as follows.

Products / Price-Ladder Constraint: For every $v_{i,j} \in V$ we have a product $u_{i,j}$, thus, the number of products in our instance is $|\mathcal{U}| = n$. The price-ladder is defined as

$$p(u_{0,0}) \leq p(u_{0,1}) \leq \cdots \leq p(u_{0,|V_0|-1}) \leq p(u_{1,0}) \leq p(u_{1,1}) \leq \cdots \leq p(u_{\Delta,|V_{\Delta}|-1}).$$

Let $\mu = 4(\Delta + 1)$ and $\gamma = \mu^{-\Delta-1}/n$. For every product $u_{i,j}$ we define a corresponding threshold

$$p_{i,j} = \mu^{i-\Delta} + j\gamma.$$

Observe that thresholds are arranged according to the price-ladder constraint and differ from each other by at least γ.

Consumers: For every $v_{i,j} \in V$ define a collection $\mathcal{C}_{i,j} = \{c^t_{i,j} \,|\, t = 0, \ldots, \mu^{\Delta-i} - 1\}$ of identical consumers with budgets $b(c^t_{i,j}, u_{i,j}) = p_{i,j}$ and $b(c^t_{i,j}, u_{k,\ell}) = p_{k,\ell}$ for all k, ℓ with $v_{k,\ell} \in \mathcal{V}(v_{i,j})$. The total number of consumers in the instance is

$$\begin{aligned} |\mathcal{C}| &\leq \sum_{i=0}^{\Delta} \sum_{j=0}^{n-1} (\mu^{\Delta-i} - 1) = n\big(\sum_{i=0}^{\Delta} \mu^i - \Delta\big) \\ &= n\big(\frac{\mu^{\Delta+1} - 1}{\mu - 1} - \Delta\big) = \mathcal{O}\big(n 4^{f(n)} f(n)^{f(n)}\big). \end{aligned}$$

Each consumer has $\mathcal{O}(f(n))$ non-zero budgets, each of which can be stored in space $\mathcal{O}(f(n))$. Thus, the total size of the instance is $\mathcal{O}(nf(n)^2 4^{f(n)} f(n)^{f(n)})$.

In analogy to the coloring of G we denote the set of all consumers as $\mathcal{C} = \mathcal{C}_0 \cup \ldots \cup \mathcal{C}_\Delta$, where $\mathcal{C}_i = \bigcup_j \mathcal{C}_{i,j}$. Note, that all budgets are consistent with the thresholds we just defined. The complete construction is illustrated in Figure 2.1.

Soundness: Let $r_{\min}(p^*)$ denote the revenue made by an optimal price assignment on the above instance. We first argue that this defines an upper bound on the size of a maximum independent set in G, i.e., $r_{\min}(p^*) \geq \alpha(G)$. Given an independent set V' of G, we can define a price assignment p as follows. If $v_{i,j} \in V'$ set $p(u_{i,j}) = p_{i,j}$, else set $p(u_{i,j}) = p_{i,j} + \gamma$. Since the $p_{i,j}$'s differ by at least γ, this assignment does not violate the price-ladder constraint.

Now consider $v_{i,j} \in V'$ and the corresponding consumers $\mathcal{C}_{i,j}$. Since $v_{k,\ell} \notin V'$ for all $v_{k,\ell} \in \mathcal{V}(v_{i,j})$, each consumer $c^t_{i,j}$ can afford to buy product $u_{i,j}$ at its threshold price $p_{i,j}$, while the prices of all products $u_{k,\ell}$ are above their thresholds and, thus, exceed the consumers' respective budgets. Hence, $u_{i,j}$ is indeed the product with smallest price that any $c^t_{i,j}$ can afford. It follows that the overall revenue from consumers $\mathcal{C}_{i,j}$ is at least

$$|\mathcal{C}_{i,j}| \cdot p_{i,j} = \mu^{\Delta-i} \left(\mu^{i-\Delta} + j\gamma\right) \geq 1.$$

Thus, price assignment p results in revenue of at least $|V'|$ and we conclude that $r_{\min}(p^*) \geq \alpha(G)$.

Completeness: Assume now that our approximation algorithm returns a price assignment p. By $r_{\min}(p)$ we refer to the overall revenue of this price assignment, $r_{\min}(p \,|\, \mathcal{C}_{i,j})$ and $r_{\min}(p \,|\, c^t_{i,j})$ denote the revenue made by sales to consumers in $\mathcal{C}_{i,j}$ and to $c^t_{i,j}$ alone, respectively. First observe that w.l.o.g. the price of each product $u_{i,j}$ is either $p_{i,j}$ or $p_{i,j} + \gamma$. To see this, note, that as long as this is not the case there is always a price that we can increase up to $p_{i,j}$ or decrease down to $p_{i,j} + \gamma$ without decreasing the overall revenue. Define

$$\mathcal{C}^+ = \left\{c^t_{i,j} \,|\, r_{\min}(p \,|\, c^t_{i,j}) = p_{i,j}\right\}$$

as the set of consumers buying at maximum possible price and $\mathcal{C}^- = \mathcal{C} \backslash \mathcal{C}^+$. Clearly, $\mathcal{C}_{i,j} \subseteq \mathcal{C}^+$ or $\mathcal{C}_{i,j} \subseteq \mathcal{C}^-$ for all i and j, since all $c^t_{i,j}$'s budgets are identical. We want to show that a large portion of the solution's revenue is due to consumers in $\mathcal{C}^+$.

Note, that a consumer $c^t_{i,j} \in \mathcal{C}^-$ buys at price at most $p_{i-1,|V_i|-1}$. Thus, we have:

$$\begin{aligned} r_{\min}(p\,|\,\mathcal{C}^-) &= \sum_{\mathcal{C}_{i,j}\subseteq\mathcal{C}^-} r_{\min}(\mathcal{C}_{i,j}) \leq \sum_{\mathcal{C}_{i,j}\subseteq\mathcal{C}^-} |\mathcal{C}_{i,j}| \cdot p_{i-1,|V_i|-1} \\ &\leq \sum_{\mathcal{C}_{i,j}\subseteq\mathcal{C}^-} \mu^{\Delta-i}(\mu^{i-1-\Delta} + n\gamma) \leq \sum_{\mathcal{C}_{i,j}\subseteq\mathcal{C}^-} \mu^{-1} + \mu^{-1} \\ &= \sum_{\mathcal{C}_{i,j}\subseteq\mathcal{C}^-} \frac{1}{2(\Delta+1)} \leq \frac{n}{2(\Delta+1)} \end{aligned}$$

On the other hand, we obtain an independent set if we pick all vertices of the same color in G and, thus, $\alpha(G) \geq n/(\Delta+1)$. Consequently, by the arguments used for completeness of the construction above, it is straightforward to construct a price assignment resulting in revenue $n/(\Delta+1)$. It follows that we may assume w.l.o.g. that $r_{\min}(p) \geq n/(\Delta+1)$ and, thus,

$$r_{\min}(p\,|\,\mathcal{C}^+) = r_{\min}(p) - r_{\min}(p\,|\,\mathcal{C}^-) \geq \frac{1}{2} r_{\min}(p).$$

Define $V' = \{v_{i,j}\,|\,\mathcal{C}_{i,j} \subseteq \mathcal{C}^+\}$. Let $v_{i,j} \in V'$ and consider the corresponding consumers $\mathcal{C}_{i,j} \subseteq \mathcal{C}^+$. The revenue made by sales to consumers in $\mathcal{C}_{i,j}$ is

$$|\mathcal{C}_{i,j}| \cdot p_{i,j} = \mu^{\Delta-i}\left(\mu^{i-\Delta} + j\gamma\right) \leq 1 + \mu^{-i-1} \leq 2.$$

We conclude that

$$|V'| = |\{\mathcal{C}_{i,j}\,|\,\mathcal{C}_{i,j} \subseteq \mathcal{C}^+\}| \geq \frac{1}{2} r_{\min}(p\,|\,\mathcal{C}^+).$$

Finally, observe that V' is indeed a feasible independent set in G. To see this, consider $v_{i,j} \in V'$ and let $v_{k,\ell}$ be an adjacent vertex. If $k < i$, the fact that consumers $\mathcal{C}_{i,j}$ buy $u_{i,j}$ at price $p_{i,j}$ implies that the price of $u_{k,\ell}$ is strictly larger than its threshold $p_{k,\ell}$. It follows that $\mathcal{C}_{k,\ell} \nsubseteq \mathcal{C}^+$ and, thus, $v_{k,\ell} \notin V'$. If $k > i$, consumers $\mathcal{C}_{k,\ell}$ can afford to buy product $u_{i,j}$ at price $p_{i,j}$ and again $\mathcal{C}_{k,\ell} \nsubseteq \mathcal{C}^+$.

Finally, observe that the proof goes through without the price-ladder constraint, as well, in which case we can w.l.o.g. assume that each product's price is equal to either its threshold price or $+\infty$. □

Theorem 2.3.7 yields several tight hardness results for unit-demand pricing in the min-buying model. Lower bounds for approximation guarantees expressed in terms of different parameters under different complexity theoretic assumptions are obtained by choosing function $f(n)$ in Theorem 2.3.7 appropriately. Corollary 2.3.8 establishes a near-tight lower bound for the approximation guarantee of the single-price algorithm expressed in terms of the number of consumer samples.

Corollary 2.3.8. UDP($\mathcal{C}$)-MIN *and* UDP($\mathcal{C}$)-MIN-PL *are not approximable in polynomial time within* $O(\log^\varepsilon m)$ *for some* $\varepsilon > 0$*, unless* NP $\subseteq$ DTIME($n^{\mathcal{O}(\log\log n)}$).

Proof: Choose $f(n) = \log n$. Then $\log^\gamma m = \mathcal{O}(\log^\varepsilon n)$ for any $\gamma \leq \varepsilon/2$ and we obtain instances of size at most $\mathcal{O}(n^{\mathcal{O}(\log\log n)})$. □

Under a somewhat stronger complexity theoretic assumption we are able to establish a lower bound corresponding to the alternative analysis of the single-price algorithm in Theorem 2.2.2, as well.

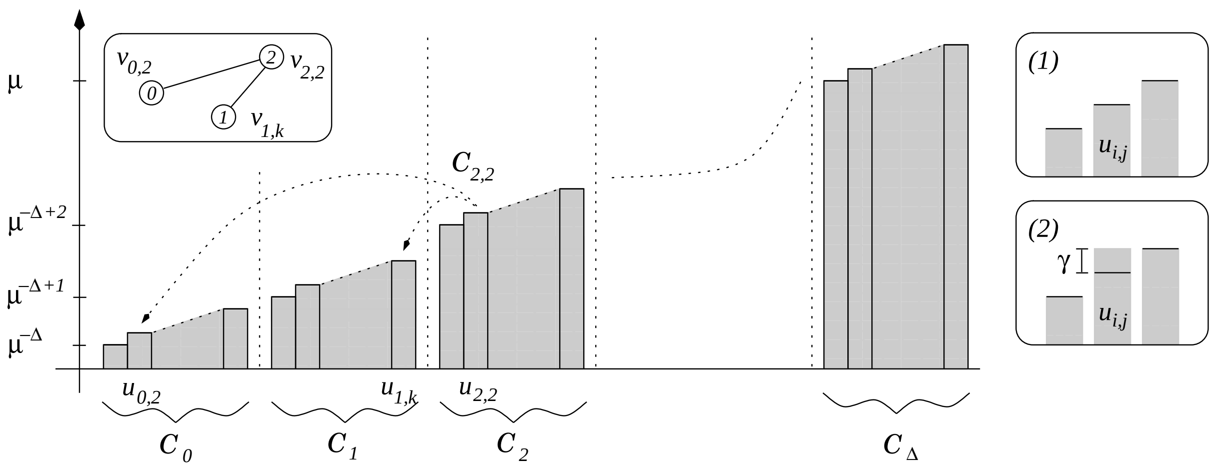


Figure 2.1: Products are arranged in blocks according to the $(\Delta + 1)$-coloring of G, thresholds in block i are roughly $\mu^{-\Delta+i}$. The additional offset γ allows setting prices according to the price-ladder. Consumers $\mathcal{C}_{2,2}$ belonging to vertex $v_{2,2}$ have non-zero budgets for $u_{2,2}$ and products in blocks with lower numbers corresponding to adjacent vertices. Cases on the right illustrate how price $p(u_{i,j})$ is set to indicate that $v_{i,j} \in V'$ (1), or $v_{i,j} \notin V'$ (2).

Corollary 2.3.9. UDP($\mathcal{C}$)-MIN *and* UDP($\mathcal{C}$)-MIN-PL *are not approximable in polynomial time within* $\mathcal{O}(n^{\varepsilon})$ *for some* $\varepsilon > 0$, *unless* NP $\subseteq \bigcap_{\delta>0}$ DTIME($2^{\mathcal{O}(n^{\delta})}$).

Proof: We choose $f(n) = n^{\delta}$ for arbitrarily small $\delta > 0$. We obtain instances of size $\mathcal{O}(2^{\mathcal{O}(n^{\gamma})})$ for any $\gamma > \delta$ that are hard to approximate within $n^{\varepsilon\delta}$ for some $\varepsilon > 0$ □

It is natural to ask whether better approximation guarantees can be achieved on sparse problem instances, e.g., on those in which every consumer is interested in only a small number of products. Applying the reduction in Theorem 2.3.7 to IS in constant degree graphs yields lower bounds for this type of approximation. This also proves that both general UDP($\mathcal{C}$)-MIN and UDP($\mathcal{C}$)-MIN-PL are not in APX under the standard assumption of P $\neq$ NP.

Corollary 2.3.10. *For every constant* $\ell \geq 3$, UDP($\mathcal{C}$)-MIN *and* UDP($\mathcal{C}$)-MIN-PL *with at most* ℓ *non-zero budgets per consumer are not approximable in polynomial time within* ℓ^{ε} *for some* $\varepsilon > 0$, *unless* P = NP.

Sketch of Proof: The version of Theorem 2.3.5 in [AFWZ95] states that there exists $\varepsilon > 0$, such that for every constant $\Delta \geq 3$ IS in graphs of degree at most Δ is NP-hard to approximate within a factor of Δ^{ε}. We construct an instance of UDP($\mathcal{C}$)-MIN as in the proof of Theorem 2.3.7 with parameters $\mu = \Delta^k$ and $\gamma = 1/(n\Delta^{k-1})$ for some $k \in \mathbb{N}$ to be determined later. We obtain an instance of polynomial size with at most Δ non-zero budgets per consumer. Given any price assignment of revenue $r_{\min}(p)$ on this instance the same calculations as before yield $r_{\min}(p \,|\, \mathcal{C}^+) \geq (1 + 1/\Delta^{k-1})^{-1} r_{\min}(p)$ and we can construct a corresponding independent set of size at least $(1 + 1/\Delta^{k-1})^{-1} r_{\min}(p \,|\, \mathcal{C}^+)$. Now fix any $0 < \delta < \varepsilon$. By choosing constant $k \in \mathbb{N}$ sufficiently large we have $1 + 1/\Delta^{k-1} \leq \Delta^{\delta/2}$ for arbitrary Δ and we obtain inapproximability of UDP($\mathcal{C}$)-MIN within a factor of

$$\left(1+\frac{1}{\Delta^{k-1}}\right)^{-2}\Delta^{\varepsilon}\geq\Delta^{\varepsilon-\delta},$$

which finishes the proof. □

We point out that constant ε in Corollary 2.3.10 is essentially the same as in the corresponding hardness result for IS in [AFWZ95] and, thus, our approximation preserving reduction is loss-free in this respect.

Finally, let us have a brief look at the distribution-based version of unit-demand pricing. In this setting, we can encode the collection $\mathcal{C}_{i,j}$ of consumers by a single consumer $c^t_{i,j}$ to which we assign probability $|\mathcal{C}_{i,j}|/|\mathcal{C}|$ in the finite support distribution. Thus, we can encode general IS in pricing instances of polynomial size and get tight hardness of approximation results based on [Zuc06].

Corollary 2.3.11. *Distribution-based* UDP($\mathcal{D}$)-MIN *and* UDP($\mathcal{D}$)-MIN-PL *are not approximable in polynomial time within $\mathcal{O}(n^{1-\varepsilon})$ for any $\varepsilon > 0$, unless* P = NP.

2.4 An $\mathcal{O}(\ell)$-Approximation

The last part of this chapter is again devoted to some algorithmic question. We will briefly sketch the application of a simple randomized algorithm, which was originally designed by Balcan and Blum [BB06] for the single-minded pricing problem, to UDP($\mathcal{C}$)-MIN, thereby providing the missing upper bound corresponding to Corollary 2.3.10. Let us assume that we are given a UDP($\mathcal{C}$)-MIN instance in which each consumer has at most ℓ non-zero budgets.

Algorithm 2: The random-partitioning algorithm for UDP($\mathcal{C}$)-MIN with at most ℓ non-zero budgets per consumer.

1. Define $\mathcal{P} \subseteq \mathcal{U}$ by placing each product in $\mathcal{P}$ independently with probability $1/\ell$. Set prices of products not in $\mathcal{P}$ to $+\infty$.
2. Define $\mathcal{C}' \subseteq \mathcal{C}$ as the set of consumers with exactly one non-zero budget for a product in $\mathcal{P}$.
3. Compute separately for each product in $\mathcal{P}$ its optimal price with respect to consumers in $\mathcal{C}'$.

The *random-partitioning algorithm* (Algorithm 2) is based on the idea of partitioning products into $\mathcal{U} = \mathcal{P} \cup \mathcal{R}$, where each product is placed in $\mathcal{P}$ with probability $1/\ell$ and in $\mathcal{R}$ with probability $1 - 1/\ell$. Let $\mathcal{C}'$ be the set of those consumers that have a non-zero budget for exactly one of the products in $\mathcal{P}$. Consider some product $u \in \mathcal{U}$ and assume that consumer $c \in \mathcal{C}$ buys u under the optimal price assignment. A simple calculation yields that

$$\Pr\big(u\in\mathcal{P}\text{ and }c\in\mathcal{C}'\big)\geq\frac{1}{\ell}\left(1-\frac{1}{\ell}\right)^{\ell-1}\geq\frac{1}{\mathrm{e}\ell}.$$

We set the prices of all products in $\mathcal{R}$ to $+\infty$ and compute optimal prices for products in $\mathcal{P}$ with respect to consumers in $\mathcal{C}'$. This is possible, because after the partitioning we have to take into account only a single non-zero budget per consumer. It is then straightforward to argue that the expected revenue from selling each product in $\mathcal{P}$ to consumers in $\mathcal{C}'$ is at least the revenue made by selling it at the price it is assigned in the optimal solution to these consumers. Finally, we mention that the algorithm can be derandomized by the method of small sample spaces [EGL+98]. This yields the following Theorem, which is essentially due to [BB06].

Theorem 2.4.1. *The random-sampling algorithm computes an $\mathcal{O}(\ell)$-approximation with respect to optimal revenue for* UDP($\mathcal{C}$)-MIN *with at most ℓ non-zero budgets per consumer.*

We mention that the techniques used above cannot be applied when a price-ladder constraint is given, since it is essential that prices of products in $\mathcal{R}$ are set to values strictly above the budgets of consumers in $\mathcal{C}'$. Doing this in the presence of a price-ladder constraint might make it impossible to assign optimal prices to the products in $\mathcal{P}$.

2.5 Literature

The unit-demand min-buying model was first proposed in [Rus03] and [GRR06]. Here, consumers were restricted to have the same budget for all products they are interested in (the uniform-budget case, see Chapter 5). In combination with a price-ladder constraint this problem variation can be solved in polynomial time by a simple dynamic programming approach. The problem version allowing consumers to have different budgets for different products was first considered in [AFMZ04], where among other results the single-price algorithm is shown to obtain logarithmic approximation guarantees for general sampling-based unit-demand pricing.

Hardness of approximation of the independent set problem in graphs of degree 3, the base problem of our reduction, stems from the PCP theorem [ALM+98] and [PY91], where the problem is shown to be APX-hard. The concept of randomized graph products is implicit in [Blu91] and made explicit in [BS92]. The derandomized graph product is found in [AFWZ95], where inapproximability results for the independent set problem in constant degree graphs are proven. For an overview of the construction of expander graphs in general and Ramanujan graphs specifically the reader is referred to [HLW06].

The random-sampling algorithm for bounded size consumers is found in [BB06], where its approximation guarantee is analyzed for the single-minded pricing problem. The new results in this chapter have been published in [BK07].

3 The Other End of the Chart: The Max-Buying Model

In the previous chapter we have seen that unit-demand pricing does not allow approximation guarantees essentially beyond the non-constant ratio obtained by the simple single-price algorithm when consumers act according to the min-buying selection rule, i.e., if consumers always choose to buy the least expensive alternative available to them. In this chapter we will have a look at a different selection rule, which can be considered as the counterpart to min-buying.

In the *max-buying* model, each consumer selects the most expensive among the products with prices not exceeding her respective budgets. At first glance, it might appear that max-buying is not a very natural model to consider, in the sense that consumers rarely tend to choose the most expensive alternative among substitutable products and, thus, it poorly models rational consumer behavior. However, there are a number of good reasons to have a closer look at pricing in the max-buying model. Firstly, as we shall see in Chapter 4, under quite reasonable assumptions max-buying is in fact equivalent to the much more realistic rank-buying model if we are given a price-ladder constraint. Thus, some of the results are directly applicable to this important model, as well. Secondly, unit-demand pricing in the max-buying model exhibits a problem structure that is quite different from other selection rules and allows the successful application of various well-studied algorithmic techniques. Consequently, understanding the max-buying model might be an important step towards the design of selection rules that allow us to model rational consumer behavior while still being computationally tractable even in the no-price-ladder case. Thirdly, the max-buying model allows us to extend algorithmic results even to incorporate the additional complication of limited product supply, which adds another dimension to the pricing problem.

The rest of this chapter is organized as follows. Section 3.1 gives a formal definition of the max-buying model and pricing with limited product supply. Section 3.2 proves hardness of approximation results that hold even for instances with small product supply. In Section 3.3 we analyze a generic local search algorithm and prove that it achieves approximation factor 2. Section 3.4 considers the case of a price-ladder constraint, sketches the existence of a PTAS and presents a proof of strong NP-hardness. Section 3.5 extends the analysis of the local search algorithm to bound the price of anarchy in a related pricing game that is obtained if the price of each product is fixed by an individual agent aiming to maximize her personal revenue.

3.1 Preliminaries

In the standard unlimited-supply setting the definition of max-buying is completely analogous to Definition 2.1.1 in Section 2.1.

Definition 3.1.1. *Given products $\mathcal{U}$, $|\mathcal{U}| = n$, and consumer samples $\mathcal{C}$, $|\mathcal{C}| = m$, consisting of budgets $b(c,u) \in \mathbb{R}_0^+$,* $\textsc{Udp}(\mathcal{C})$-$\textsc{Max}$ *asks for a price assignment $p : \mathcal{U} \to \mathbb{R}_0^+$ maximizing*

$$r_{\max}(p) = \sum_{c \in \mathcal{A}(p)} \max\{p(u) \mid u \in A_c(p)\}.$$

UDP($\mathcal{C}$)-MAX-PL *asks for a revenue maximizing price assignment satisfying a given price-ladder constraint* π.

In the *limited-supply setting*, we assume that product $u \in \mathcal{U}$ is available in supply $s(u) \in \mathbb{N}$. In this situation, specifying the product prices is no longer sufficient, and we additionally have to come up with some allocation $a : \mathcal{C} \to \mathcal{U}$ of the products among the consumers willing to buy. We have some freedom of choice about which kinds of allocations to allow. We will say that allocation a is *feasible* if $a(c) \in A_c(p) \cup \{\varnothing\}$ for all $c \in \mathcal{C}$ and $|a^{-1}(u)| \leq s(u)$ for all $u \in \mathcal{U}$, i.e., if each consumer can afford her allocated product and the allocation of no product exceeds its supply. Note, that allocation a is not required to obey consumer preferences as expressed through the selection rule. To capture the selection rule, as well, we say allocation a is *strictly feasible*, if $a(c) \in \{s(c,p), \varnothing\}$, i.e., if every consumer is allocated the product specified by the selection rule or nothing at all.

Clearly, we could define further variations of feasible allocations, e.g., allocating to each consumer the product most desirable according to the selection rule among those which are still available. However, in the case of max-buying most of these variations are essentially captured by the notion of feasibility as defined above, since the revenue maximizing allocation will always try to sell the most expensive product still available.

Given prices p, it is not difficult to compute the revenue maximizing allocation. We think of consumers and products as vertices on opposite sides of a bipartite graph. A vertex corresponding to consumer c is adjacent to all vertices corresponding to her affordable products $A_c(p)$. An edge is assigned the price of the incident product vertex as weight. Finally, the vertex corresponding to product u gets degree constraint $s(u)$, every consumer vertex gets degree constraint 1. The revenue maximizing feasible allocation is equivalent to a maximum weight b-matching in this graph, which can be found in polynomial time [CCPS98]. If we are interested in a strictly feasible allocation, consumer c's vertex is connected only to the vertex representing her desired product $s(c,p)$. Given prices p and allocation a we denote by $r(p,a)$ the overall revenue.

Proposition 3.1.2. *Given fixed prices p for products $\mathcal{U}$, a (strictly) feasible allocation $a : \mathcal{C} \to \mathcal{U}$ to consumers in $\mathcal{C}$ maximizing $r(p,a)$ can be found in polynomial time.*

3.2 Hardness of Approximation

In [AFMZ04] it is shown how to obtain a 1.59-approximation for UDP($\mathcal{C}$)-MAX with unlimited supply based on randomized LP-rounding techniques. The problem is also proven to be APX-hard in general. We want to investigate here specifically the effect of limited product supply and, more precisely, answer the question at what maximum supply hardness of approximation kicks in.

In the case of *unit-supply* we assume that there is exactly one copy of each product available, thus, $s(u) = 1$ for all $u \in \mathcal{U}$. As we have argued before prices can w.l.o.g. be chosen from the set of distinct budget values. Hence, in the unit-supply case the price of a product is determined solely by the budget of the consumer who buys it. But then every fixed allocation implies a corresponding price assignment and the problem reduces to finding the optimal allocation. This, however, is equivalent to solving a weighted matching problem in a bipartite graph, with vertices on opposite sides of the bipartition representing consumers and products, respectively. Non-zero budgets $b(c,u)$ are represented as edges with weight $b(c,u)$ connecting the vertices of consumer c and product u. We have thus obtained the following result.

Theorem 3.2.1. UDP($\mathcal{C}$)-MAX *with unit-supply can be solved in polynomial time.*

The reduction used to prove APX-hardness in [AFMZ04] creates problem instances with maximum supply that is linear in the number of products. It is then a natural question to ask how the problem complexity behaves in between these extremes, i.e., for maximum supply that is larger than 1 but still small compared to the size of the problem instance. Surprisingly, it turns out that even increasing maximum supply to only 2 is sufficient to make the problem APX-hard.

Theorem 3.2.2. UDP($\mathcal{C}$)-MAX *with limited supply* 2 *or larger is* APX-*hard.*

Proof: We show an approximation preserving reduction from MAXCUT. It is known that MAXCUT is APX-hard even for graphs with maximum degree 3 (see, e.g., [ACG+99]). Let $G = (V,E)$ have such bounded degree. We transform G into an UDP($\mathcal{C}$)-MAX instance as follows. For each vertex $v \in V$ we define products $u_v^0, \ldots, u_v^5$ available in supply $s(u_v^0) = s(u_v^2) = s(u_v^4) = 2$, $s(u_v^1) = s(u_v^3) = s(u_v^5) = 1$ and consumers $c_v^0, \ldots, c_v^5$ with budget values $b(c_v^i, u_v^i) = b(c_v^i, e_v^{i+1}) = 1$ for $i \in \{0,2,4\}$, $b(c_v^i, e_v^i) = b(c_v^i, e_v^{i+1 \mod 6}) = 2$ for $i \in \{1,3,5\}$. Budgets that are not specified are assumed to be 0. Each edge $e = \{v,w\} \in E$ can now be associated with unique products u_v^i and u_w^j, where $i,j \in \{0,2,4\}$ and every product is associated with at most one edge. For edge e we define 2 consumers c_e^0 and c_e^1 with budgets $b(c_e^0, u_v^i) = b(c_e^0, e_w^j) = 1$, $b(c_e^1, u_v^i) = b(c_e^1, e_w^j) = 2$. This construction is depicted in Figure 3.1.

We start by stating some facts about the solution that an approximation algorithm for UDP($\mathcal{C}$)-MAX might return on this instance. First, we observe that we can w.l.o.g. assume that all prices in this solution are from $\{1,2\}$, since prices above 2 cannot result in any revenue and prices below 2 can always be increased up to the next budget value. The second important observation relates price assignments to feasible cuts of the original graph.

Fact 3.2.3. *For any price assignment* p *on the above* UDP($\mathcal{C}$)-MAX *instance we can assume w.l.o.g. that* $p(u_v^0) = p(u_v^2) = p(u_v^4)$*,* $p(u_v^1) = p(u_v^3) = p(u_v^5)$ *and* $p(u_v^0) \neq p(u_v^1)$ *for all* $v \in V$*.*

We show how any solution in which the above assumption does not hold can be turned into a solution of no smaller revenue, such that it does. We first consider the case that products u_v^0, u_v^2 and u_v^4 have not been assigned identical prices. For reasons of symmetry it is sufficient to consider the situation that product u_v^0 has been assigned the wrong price.

First, assume that $p(u_v^0) = 2$, $p(u_v^2) = p(u_v^4) = 1$. In this situation, if $p(u_v^1) = 2$, consumer c_v^0 currently cannot afford to buy any product, resulting in revenue 0 from this consumer. If $p(u_v^1) = 1$, then consumer c_v^1 currently buys at price at most 1. In both cases, the revenue generated by consumers $c_v^0, \ldots, c_v^5$ is at most 8. By setting $p(u_v^0) = p(u_v^2) = p(u_v^4) = 1$, $p(u_v^1) = p(u_v^3) = p(u_v^5) = 2$ and $a(c_v^i) = u_v^i$ for all i this revenue increases to 9. On the other hand, if product u_v^0 is associated with some edge e, only 1 consumer from $\{c_e^0, c_e^1\}$ can afford product u_v^0 at price 2 and, thus, might be buying it. Revenue from this consumer decreases by no more than 1. Hence, we have transformed our solution without decreasing the overall revenue.

For the second case, let $p(u_v^0) = 1$, $p(u_v^2) = p(u_v^4) = 2$. If $p(u_v^5) = 2$, consumer c_v^4 cannot afford any product. If $p(u_v^5) = 1$, consumer c_v^5 buys at price at most 1. Again setting $p(u_v^0) = p(u_v^2) = p(u_v^4) = 2$,

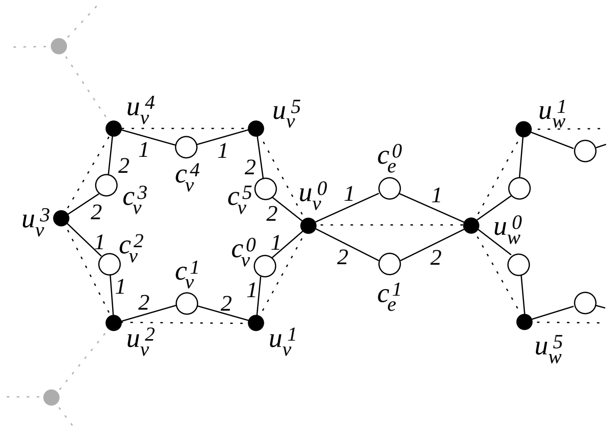


Figure 3.1: Construction from the proof of Theorem 3.2.2. Consumers are depicted as circles, products as points. Edges between consumers and products are labelled with the respective non-zero budgets.

$p(u_v^1) = p(u_v^3) = p(u_v^5) = 1$ and $a(c_v^i) = u_v^{i+1 \mod 6}$ makes overall revenue from consumers $c_v^0, \ldots, c_v^5$ increase by 1. On consumers $\{c_e^0, c_e^1\}$ revenue decreases by at most 1, because consumer c_e^1 can still buy a product at price 2 after $p(u_v^0)$ is changed.

Finally, assuming that products u_v^0, u_v^2 and u_v^4 have been assigned the same price, it is straightforward to argue that overall revenue becomes maximal when products u_v^1, u_v^3, u_v^5 are assigned identical prices not equal to $p(u_v^0)$. This proves Fact 3.2.3.

We now argue how any small constant factor approximation on the constructed problem instance yields a corresponding approximation for the MAXCUT problem. As we have seen we obtain solutions with prices in $\{1, 2\}$ as described in Fact 3.2.3 and a corresponding allocation a for all $v \in V$. Thus, overall revenue from consumers $c_v^0, \ldots, c_v^5$ is exactly 9 for all $v \in V$. For consumers $\{c_e^0, c_e^1\}$ belonging to some edge $e = \{v, w\}$ it is simple to find the optimal allocation given prices $p(u_v^i)$, $p(u_w^j)$ of the corresponding products. If $p(u_v^i) = p(u_w^j) = 1$ then we can set $a(c_e^0) = u_v^i$, $a(c_e^1) = u_w^j$. If $p(u_v^i) = p(u_w^j) = 2$ then we let $a(c_e^0) = \varnothing$, $a(c_e^1) = u_v^i$. If $p(u_v^i) = 1$, $p(u_w^j) = 2$ we define $a(c_e^0) = u_v^i$, $a(c_e^1) = u_w^j$. Thus, total revenue from consumers c_e^0 and c_e^1 is 2 if $p(u_v^i) = p(u_w^j)$ and 3 if $p(u_v^i) \neq p(u_w^j)$. We can then write the value of any such solution to UDP($\mathcal{C}$)-MAX as $9n+2m+c$, where $n = |V|$, $m = |E|$, and c is the number of edges $\{v, w\}$ such that $p(u_v^0) \neq p(u_w^0)$. Given this solution we can immediately define a cut (S, T) of size c in G by setting $S = \{v \mid p(u_v^0) = 1\}$, $T = V \setminus S$. Hence, the optimal solution on our pricing instance has value $9n + 2m + c^*$, where c^* is the size of a maximum cut in G. Assume now that we can obtain a $(1 - \varepsilon)$-approximation to the pricing problem. By $n \leq m$ (assuming G is connected and not a tree) and $c^* \geq m/2$ we have

$$(1 - \varepsilon) \leq \frac{9n + 2m + c}{9n + 2m + c^*} \leq \frac{22c^* + c}{23c^*}$$

and, thus, $c/c^* \geq (1 - 23\varepsilon)$. Choosing ε appropriately small yields any arbitrarily small constant approximation ratio for MAXCUT. □

3.3 A Local Search Algorithm

We proceed by analyzing the approximation guarantee of a generic local search approach to UDP($\mathcal{C}$)-MAX with limited supply. For a given price assignment p let $[p \,|\, p(u) = p']$ refer to the price assignment obtained by changing the price of u to p'. The *local-search algorithm* (Algorithm 3) tries to improve its current price assignment p by checking $[p \,|\, p(u) = p']$ for all $u \in \mathcal{U}$, $p' \neq p(u)$, and terminates with a solution that cannot be improved by changing a single price.

Algorithm 3: The local-search algorithm for UDP($\mathcal{C}$)-MAX with limited product supply.

Initialize p arbitrarily and compute the optimal (strictly) feasible allocation a.
while *there exists product* u *and price* $p' \neq p(u)$ *such that*

$$r(p, a) < r([p \,|\, p(u) = p'], a'),$$

where a' *is the optimal allocation given prices* $[p \,|\, p(u) = p']$ **do**
  Set $p(u) = p'$.

Aggarwal et al. [AFMZ04] show how to obtain a 1.59-approximation for unlimited-supply UDP($\mathcal{C}$)-MAX based on an LP-relaxation and randomized rounding. Due to the probabilistic nature of their algorithm it appears rather difficult to extend it to general limited-supply scenarios. Apart from this, the algorithm is based on a somewhat problem-specific LP-formulation. Local search, on the other hand, appears to be a quite natural approach to a wide range of pricing problems. Seen in this light, we provide first evidence that this approach might indeed be promising also for more practical problems.

We next show that the total revenue generated by the locally optimal solution returned by the local-search algorithm lies within a factor of 2 off the globally optimal solution's value.

Theorem 3.3.1. *Let p be the price assignment returned by the local-search algorithm, p^* an optimal price assignment and a, a^* the respective optimal (strictly) feasible allocations. Then $r(p^*, a^*)/r(p, a) \leq 2$ and, thus, the local-search algorithm achieves approximation ratio 2 for* UDP($\mathcal{C}$)-MAX *with limited or unlimited supply. Furthermore, this bound is tight.*

Proof: Consider price assignment p and allocation a returned by the algorithm. We define $C_u = (a^*)^{-1}(u)$, $L_u = \{c \in C_u \,|\, p(a(c)) < p^*(u)\}$ and $r_u = p(u)|a^{-1}(u)|$, i.e., C_u refers to the set of consumers buying u in an optimal solution, L_u is the subset of these consumers that buy at a price below $p^*(u)$ in the solution returned by the local-search algorithm and r_u denoted the revenue due to product u in this solution. Furthermore, we let

$$\Delta_u = \sum_{c \in L_u} \Big(p^*(u) - p(a(c)) \Big)$$

refer to the loss of revenue compared to the optimal solution incurred by consumers in C_u. Changing price $p(u)$ to $p^*(u)$ (or leaving it as it is in case it should happen to be just $p^*(u)$) defines price assignment $p' = [p \,|\, p(u) = p^*(u)]$ and corresponding allocation a'. Since we do not know what a' should look like we define an alternative allocation a'' as follows. First, we set $a''(c) = \varnothing$ for all consumers c with $a(c) = u$. We then set $a''(c) = u$ for all $c \in L_u$. For all other consumers we do not change allocation a

and let $a''(c) = a(c)$. First observe that allocation a'' does not allocate more copies of any product than there are available, since $|L_u| \leq |C_u| \leq s(u)$ and no product besides u can be sold to more consumers than in a. Then note that a'' is (strictly) feasible if this was true for a, since u is now the most expensive affordable product for all consumers in L_u. It clearly holds that $r(p', a') \geq r(p', a'')$ by the optimality of a'. We observe that

$$\begin{aligned} r(p',a') - r(p,a) &\geq r(p',a'') - r(p,a) \\ &= \sum_{c \notin L_u \cup a^{-1}(u)} p(a(c)) + \sum_{c \in L_u} p^*(u) - \sum_{c \in \mathcal{C}} p(a(c)) \\ &\geq \sum_{c \in \mathcal{C}} p(a(c)) + \sum_{c \in L_u} (p^*(u) - p(a(c))) - \sum_{c \in a^{-1}(u)} p(a(c)) - \sum_{c \in \mathcal{C}} p(a(c)) \\ &= \Delta_u - r_u. \end{aligned}$$

By the fact that $r(p,a)$ cannot be improved by changing a single price $p(u)$ we have that $r(p',a') - r(p,a) \leq 0$ and, thus, $r_u \geq \Delta_u$. (If price $p(u)$ did not have to be changed because it was already $p^*(u)$ the same inequality follows from the optimality of allocation a.) Let now $r_u^* = p^*(u)|C_u|$ denote the revenue made by product u in the optimal solution. We can then write that

$$\begin{aligned} 2 \cdot r(p,a) &= \sum_{u \in \mathcal{U}} r_u + \sum_{c \in \mathcal{C}} p(a(c)) \\ &\geq \sum_{u \in \mathcal{U}} \big(r_u + \sum_{c \in C_u} p(a(c))\big) \\ &\geq \sum_{u \in \mathcal{U}} (r_u + r_u^* - \Delta_u) \\ &\geq \sum_{u \in \mathcal{U}} r_u^* = r(p^*, a^*). \end{aligned}$$

This completes the first part of the proof. It remains to be shown that our analysis is tight. To this end, consider a problem instance with 2 products u_1, u_2 and $k+1$ consumers $c_1, \ldots, c_{k+1}$. Consumer budgets are $b(c_1, u_1) = k$, $b(c_1, u_2) = k - \varepsilon$, $b(c_2, u_1) = 0$, $b(c_2, u_2) = \varepsilon$ and $b(c_i, u_1) = 1$, $b(c_i, u_2) = 0$ for $i = 3, \ldots, k+1$. We assume that products are available in unlimited supply. It is straightforward to verify that prices $p(u_1) = k$, $p(u_2) = \varepsilon$ are locally optimal and result in revenue $k + \varepsilon$. Prices $p(u_1) = 1$, $p(u_2) = k - \varepsilon$, however, result in overall revenue of $2k - 1 - \varepsilon$. Choosing k and ε appropriately shows that a pure local search approach cannot give any approximation ratio better than 2. □

So far, we have argued that the local-search algorithm terminates with a solution that is a 2-approximation with respect to the optimal revenue. We have not, however, argued about the algorithm's running time. In order to obtain polynomial running time, the following small change in the algorithm is fully sufficient. Instead of choosing any improving step, we need to find in each iteration the new price that will give maximum increase in revenue. This yields the following theorem.

Theorem 3.3.2. UDP($\mathcal{C}$)-MAX *with limited or unlimited supply and integral budgets can be approximated in polynomial time within a factor of* 2.

Proof: Assume that we choose in each iteration the new price that will give maximum increase in revenue. Let r be the revenue of the current solution, r^* the revenue of an optimal solution and assume that $r^* - 2r \geq \phi$. Using the same notation as in the proof of Theorem 3.3.1 there must exist a product u, such that $r_u \leq \Delta_u - \phi/n$, where n denotes the number of products in the instance. It follows that revenue increases by at least ϕ/n in each iteration and, thus, after k iterations it must be true that

$$\phi \leq r^* \left(1 - \frac{2}{n}\right)^k ,$$

since in the first iteration it holds that $\phi \leq r^*$. We assume that all budgets are integral. It follows that the overall revenue increases by at least 1 in each iteration. Now let $\ell = n \cdot \lceil \ln r^* \rceil + 1$. After ℓ iterations we have that

$$\phi \leq r^* \left(1 - \frac{2}{n}\right)^{n \cdot \ln r^*} - 1 \leq r^* \cdot \mathrm{e}^{-\ln r^*} - 1 = 0,$$

and, thus, we can terminate the algorithm after ℓ iterations with an approximation guarantee of 2. Note, that we do not need to know the value of r^*. For (weakly) polynomial running time it is sufficient to upper bound r^* by the sum of consumers' maximum budgets. □

The analysis of the local-search algorithm can easily be extended to capture scenarios in which consumers are described by various types of probability distributions, as long as these distributions can somehow be accessed efficiently. Especially, this is clearly true for finite support distributions as introduced in Section 2.1 and we obtain the following extension of Theorem 3.3.2.

Theorem 3.3.3. UDP($\mathcal{D}$)-MAX *with limited or unlimited supply and integral budgets can be approximated in polynomial time within a factor of* 2.

Finally, let us briefly discuss the reason why local search can be successfully applied to UDP($\mathcal{C}$)-MAX. The max-buying model exhibits an interesting *revenue transfer property* that separates it from the other models considered here. Assume that we set the price of a single product u to its optimal value $p^*(u)$ and a consumer c that buys u in the optimal solution does not select u. Then it must be the case that c buys another product at an even higher price and, thus, revenue that is made by selling u in the optimal solution is not lost but *transferred* to some other product u'.

3.4 Max-Buying with Price-Ladder Constraint

We now return to the case of unlimited product supply and assume that we are given a price-ladder constraint. As we point out in Chapter 4 in this situation max-buying is equivalent to a relevant variation of rank-buying. We first sketch a polynomial time approximation scheme (PTAS) in Section 3.4.1 and then present a proof of strong NP-hardness in Section 3.4.2.

3.4.1 A PTAS

We briefly sketch the PTAS for UDP($\mathcal{C}$)-MAX-PL from [AFMZ04], which is based on the following relaxation of the problem. First, we restrict ourselves to prices that are powers of $s = (k+1)^{1/k}$. We then allow each consumer to buy multiple products, as long as the prices of any two products vary by at least a factor of $(k+1)$. This yields an approximation guarantee of $(k+1)^{1/k}(1+(1/k)) = 1+(\log k/k)(1+o(1))$, which can be made $1+\varepsilon$ for any $\varepsilon > 0$ by choosing constant k sufficiently large.

The relaxed problem is solved by the following dynamic programming approach. Assume that we are given a price-ladder constraint $p(u_1) \leq \cdots \leq p(u_n)$ and that $s^{t+1} > b(c,u)$ for all $c \in \mathcal{C}$, $u \in \mathcal{U}$. Thus, it is sufficient to consider prices $s^0, \ldots, s^t$. By $F_i(x_i, \ldots, x_{i+k-1})$ we denote the maximum revenue considering only products priced at s^i or higher and x_j is the index of the first product with price s^j or higher. Now let $C(x_{i-1}, \ldots, x_{i+k-1})$ denote the number of consumers that cannot afford to buy any of the products $u_{x_j}, \ldots, u_{x_{j+1}-1}$ at price s^j for $j = i, \ldots, i+k-2$, but can afford to buy at least one of the products $u_{x_{i-1}}, \ldots, u_{x_i-1}$ at price s^{i-1}. We then define

$$G_{i-1}(x_{i-1}, \ldots, x_{i+k-1}) = F_i(x_i, \ldots, x_{i+k-1}) + s^{i-1}C(x_{i-1}, \ldots, x_{i+k-1}).$$

Now, iterating over all values of x_{i+k-1} one obtains the following recurrence for the F-values:

$$F_{i-1}(x_{i-1}, \ldots, x_{i+k-2}) = \max_{x_{i+k-2} \leq x_{i+k-1} \leq n} G_{i-1}(x_{i-1}, \ldots, x_{i+k-1}).$$

A simple calculation shows that the size of the dynamic programming table is polynomial for any fixed constant k.

Theorem 3.4.1 ([AFMZ04]). UDP($\mathcal{C}$)-MAX-PL *allows a PTAS.*

3.4.2 Strong NP-Hardness

We proceed by showing that the PTAS presented in the previous section is the best approximation result one can hope for by a proof of strong NP-hardness. The proof relies on a reduction of MAX-2SAT, where the main technical difficulty lies in encoding the problem in a way, such that prices can be set according to a predefined price-ladder. We point out that without significant modifications the proof goes through for the (more practical) rank-buying model, as well (see Chapter 4).

Theorem 3.4.2. UDP($\mathcal{C}$)-MAX-PL *with unlimited supply is strongly* NP-*hard, even if each consumer has at most* 2 *non-zero budgets.*

Proof: We show that MAX-2SAT $\leq_{\mathrm{p}}$ UDP($\mathcal{C}$)-MAX-PL. MAX-2SAT is known to be NP-hard [GJ79]. As a MAX-2SAT instance we are given a collection of disjunctive clauses $c_1, \ldots, c_m$ of length at most 2 over variables $x_1, \ldots, x_n$ and some positive integer $s \in \mathbb{N}$. We ask whether there is a truth assignment $t : \{x_1, \ldots, x_n\} \to \{0,1\}$ that simultaneously satisfies s of the clauses. Note, that w.l.o.g. we may assume that $n \leq m$, since variables that appear in only a single clause can be assigned the boolean value that satisfies this clause and then be removed from the instance. We next describe a polynomial time reduction to UDP($\mathcal{C}$)-MAX.

Variable gadgets: For every variable x_i we construct a gadget $\mathcal{V}_i$ consisting of 2 products u_i, $\overline{u}_i$ and the following collection of consumers:

- α_i^j, $j = 1, \ldots, 4m$, with budgets $b(\alpha_i^j, u_i) = 1 + \frac{2i-2}{2m^2}$ and $b(\alpha_i^j, \overline{u}_i) = 1 + \frac{2i-1}{2m^2}$.

- β_i^j, $j = 1, \ldots, 4m^3$, with budgets $b(\beta_i^j, u_i) = 1 + \frac{2i-1}{2m^2}$.
- γ_i^j, $j = 1, \ldots, 4m^3 + 4m$, with budgets $b(\gamma_i^j, \overline{u}_i) = 1 + \frac{2i}{2m^2}$.

Budgets that are not explicitly stated are assumed to be 0. By $r_{\max}(\mathcal{V}_i)$ we refer to the revenue made from sales to the above consumers. We proceed by calculating the value of $r_{\max}(\mathcal{V}_i)$ depending on prices $p(u_i)$ and $p(\overline{u}_i)$. Let $r_i^* = (4m^3 + 4m)(2 + (4i - 2)/(2m^2))$.

(1) $p(u_i) = 1 + \frac{2i-2}{2m^2}$, $p(\overline{u}_i) = 1 + \frac{2i}{2m^2}$. Consumers α_i^j and β_i^j buy u_i, γ_i^j buy $\overline{u}_i$.

$$\begin{aligned} r_{\max}(\mathcal{V}_i) &= (4m^3 + 4m)(1 + \frac{2i-2}{2m^2}) + (4m^3 + 4m)(1 + \frac{2i}{2m^2}) \\ &= (4m^3 + 4m)(2 + \frac{4i-2}{2m^2}) = r_i^*. \end{aligned}$$

(2) $p(u_i) = 1 + \frac{2i-1}{2m^2}$, $p(\overline{u}_i) = 1 + \frac{2i-1}{2m^2}$. Consumers β_i^j buy u_i, α_i^j and γ_i^j buy $\overline{u}_i$.

$$\begin{aligned} r_{\max}(\mathcal{V}_i) &= 4m^3(1 + \frac{2i-1}{2m^2}) + (4m^3 + 8m)(1 + \frac{2i-1}{2m^2}) \\ &= (4m^3 + 4m)(2 + \frac{4i-2}{2m^2}) = r_i^*. \end{aligned}$$

(3) $p(u_i) = 1 + \frac{2i-2}{2m^2}$, $p(\overline{u}_i) = 1 + \frac{2i-1}{2m^2}$. Consumers β_i^j buy u_i, α_i^j and γ_i^j buy $\overline{u}_i$.

$$\begin{aligned} r_{\max}(\mathcal{V}_i) &= 4m^3(1 + \frac{2i-2}{2m^2}) + (4m^3 + 8m)(1 + \frac{2i-1}{2m^2}) \\ &= 4m^3(2 + \frac{4i-3}{2m^2}) + 4m(2 + \frac{4i-2}{2m^2}) = r_i^* - 2m. \end{aligned}$$

(4) $p(u_i) = 1 + \frac{2i-1}{2m^2}$, $p(\overline{u}_i) = 1 + \frac{2i}{2m^2}$. Consumers β_i^j buy u_i, γ_i^j buy $\overline{u}_i$.

$$\begin{aligned} r_{\max}(\mathcal{V}_i) &= 4m^3(1 + \frac{2i-1}{2m^2}) + (4m^3 + 4m)(1 + \frac{2i}{2m^2}) \\ &= 4m^3(2 + \frac{4i-1}{2m^2}) + 4m(1 + \frac{2i}{2m^2}) \leq r_i^* - 2m. \end{aligned}$$

We observe that optimal revenue is obtained with prices set as in cases (1) and (2). If prices are set as in cases (1) and (2) we say that $\mathcal{V}_i$ is *in state* 1 or *in state* 0, respectively. In our interpretation variable gadgets in state 0 correspond to variables that are assigned the boolean value 0, variable gadgets in state 1 to variables that are assigned 1. We next describe how to encode clauses.

Clause gadgets: For every clause c_j we define a single consumer δ_j with the following budgets:

- $b(\delta_j, u_i) = 1 + \frac{2i-2}{2m^2}$, if clause c_j contains literal x_i.
- $b(\delta_j, \overline{u}_i) = 1 + \frac{2i-1}{2m^2}$, if clause c_j contains literal $\overline{x}_i$.

Again, budgets that are not explicitly stated are set to 0. We finally impose a price-ladder constraint that requires that

$$p(u_1) \leq p(\overline{u}_1) \leq p(u_2) \leq p(\overline{u}_2) \leq \cdots \leq p(u_n) \leq p(\overline{u}_n)$$

and let $r^*_{\max} = \sum_i r^*_i$. For the constructed UDP($\mathcal{C}$)-MAX instance we now ask whether there exists a price assignment p that result in overall revenue of at least $r^*_{\max} + s$ for $s \in \mathbb{N}$ as in the MAX-2SAT instance. The idea of the construction is depicted in Figure 3.2. We proceed by proving the correctness of the above reduction.

Soundness: Let t be a truth assignment satisfying s of the clauses. If $t(x_i) = 0$ we set variable gadget $\mathcal{V}_i$ to state 0, if $t(x_i) = 1$ to state 1. Clearly, our price assignment is in accordance with the price-ladder constraint. Consider a satisfied clause c_j. If c_j contains literal x_i and $t(x_i) = 1$, then the corresponding consumer δ_j can afford to buy product u_i at its price $p(u_i) = 1 + (2i-2)/(2m^2)$. On the other hand, if c_j contains $\overline{x}_i$ and $t(x_i) = 0$, then δ_j can afford $\overline{u}_i$ at price $p(\overline{u}_i) = 1 + (2i-1)/(2m^2)$. In both cases δ_j will buy some product at a price of at least 1. Thus, overall revenue is at leat $r^*_{\max} + s$.

Completeness: Let p be a price assignment resulting in revenue at least $r^*_{\max} + s$. We construct a truth assignment t that satisfies s of the clauses. We first argue that w.l.o.g. each variable gadget $\mathcal{V}_i$ is in either state 0 or state 1.

Observe first that prices $p(u_i)$ and $p(\overline{u}_i)$ can w.l.o.g. be assumed to be from the set of distinct budget values of consumers interested in these products, since, as long as this is not the case, there always exists some price that can be set to the nearest budget value without decreasing overall revenue or violating the price-ladder constraint. Assume then that $\mathcal{V}_i$ is neither in state 0 nor in state 1 and let k be the number of consumers of type δ_j buying u_i or $\overline{u}_i$. By cases (3) and (4) above the total revenue made by selling products u_i and $\overline{u}_i$ is bounded by

$$r^*_i - 2m + k(1 + \frac{2i-1}{2m^2}) \leq r^*_i - 2m + k(1 + \frac{1}{m}) < r^*_i.$$

On the other hand, setting prices as in state 0 or state 1 will give revenue at least r^*_i from selling products u_i and $\overline{u}_i$ to consumers of type α^j_i, β^j_i and γ^j_i.

We can then define the obvious truth assignment t by $t(x_i) = 0$ if $\mathcal{V}_i$ is in state 0, $t(x_i) = 1$ if $\mathcal{V}_i$ is in state 1. For every consumer δ_j that can afford to buy a product under price assignment p the corresponding clause c_j is satisfied by t. Since revenue made by sales to consumers of type α^j_i, β^j_i and γ^j_i is precisely $r^*_{\max}$, the number of consumers of type δ_j buying some product must be at least

$$\left\lceil s \cdot (1 + \frac{2m-1}{2m^2})^{-1} \right\rceil \geq \left\lceil s \cdot (1 + \frac{1}{m})^{-1} \right\rceil \geq s,$$

where we use the fact that $s \leq m$ and consumers δ_j buy at a price of at most $1 + (2m-1)/(2m^2)$. This finishes the proof. □

3.5 A Max-Buying Pricing Game

Finally, we are going to show that the analysis of the local-search algorithm from Section 3.3 can be extended to bound the *price of anarchy* [KP99], i.e., the worst case ratio between the revenue of an optimal

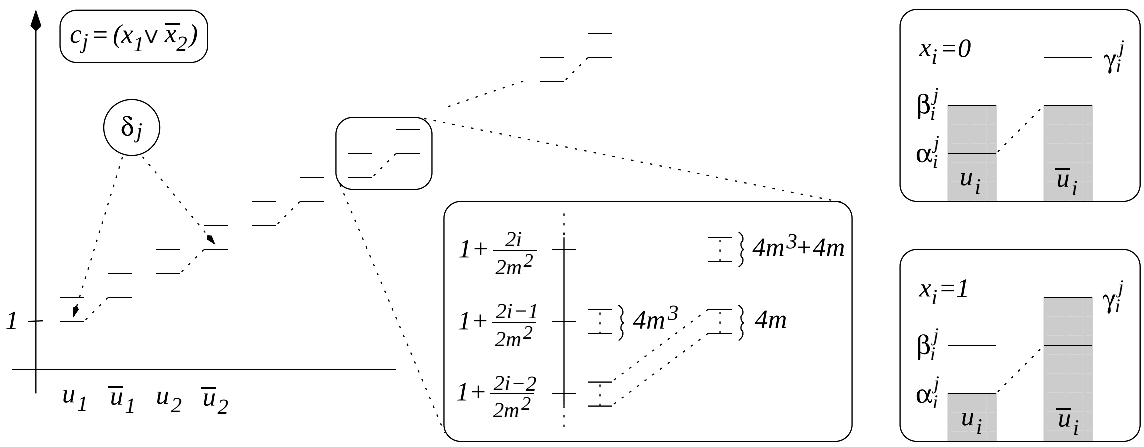


Figure 3.2: Consumers $\alpha_i^j, \beta_i^j, \gamma_i^j$ ensure that prices of u_i and $\overline{u}_i$ are always in state 0 ($x_i = 0$) or in state 1 ($x_i = 1$), both of which are consistent with the price-ladder constraint. For each clause c_j we have a single consumer δ_j with non-zero budgets for the products corresponding to the literals of c_j.

solution and any *Nash equilibrium*, in the pricing game we obtain if we let an individual player fix the price of each product. Since it can be shown that *pure* Nash equilibria do not generally exist, we will work here with the concept of *mixed* equilibria. Interestingly, the price of anarchy turns out to be 2, so in order to obtain good revenue in the max-buying scenario not even a global objective seems to be necessary. First let us introduce some notation to describe *mixed strategies*. Let $\mathcal{P} = \{1, \ldots, n\}$ be a set of players. Each player j needs to assign a price p_j to her product u_j, such as to maximize her revenue from sales to consumers $\mathcal{C}$. Allowing mixed strategies, every player defines a probability distribution P_j over a set of possible prices for her product u_j. For ease of notation we let $P = (P_1, \ldots, P_n)$, $P_{-j} = (P_1, \ldots, P_{j-1}, P_{j+1}, \ldots, P_n)$ and $(P_{-j}, P_j) = P$. Observe that we can w.l.o.g. allow only the budget values as possible prices and, thus, P_j is a discrete distribution. Since every set of fixed prices defines an optimal feasible allocation, the distributions P_j define a probability distribution also over the set of allocations. We define R_j to be the random variable that describes the revenue of player j. We can write that

$$\mathrm{E}\big[\, R_j \,\big] = \sum_{p,a} \mathrm{Pr}_P\big(p_j = p\big) \cdot \mathrm{Pr}_P\big(a \,|\, p_j = p\big) \cdot p_j |a^{-1}(u_j)|.$$

A set of strategies $P^{eq} = (P_1^{eq}, \ldots, P_n^{eq})$ are at Nash equilibrium, if for every player j we have that

$$\mathrm{E}_{P^{eq}}\big[\, R_j \,\big] \geq \mathrm{E}_{(P_{-j}^{eq}, P_j')}\big[\, R_j \,\big] \qquad \forall \quad P_j' \neq P_j^{eq},$$

i.e., if no player can increase her expected revenue by unilaterally changing her current strategy P_j^{eq}. Let prices $p_1^*, \ldots, p_n^*$ and allocation a^* be an optimal (i.e., revenue maximizing) solution to UDP($\mathcal{C}$)-MAX. Again, we let $C_j = (a^*)^{-1}(u_j)$ refer to the set of consumers that buy product u_j in this solution and define $L_j = \{c_i \in C_j \,|\, p_{a(c_i)} < p_j^*\}$, $H_j = C_j \backslash L_j$ for a fixed allocation a. For the remainder of this section it will be convenient to refer to players, their products and consumers only by their indices.

We assume that the feasible allocation of products to consumers is always chosen optimally. In order to apply arguments similar to the proof of Theorem 3.3.1 we need the following property of optimal allocations.

Lemma 3.5.1. *Consider a set of prices $p_1, \ldots, p_n$ with optimal feasible allocation a and let $|L_j| = t$. If price p_j is changed to p_j^* and we recompute the optimal allocation b we have that $|b^{-1}(j)| \geq t$.*

Proof: Throughout this proof, set L_j is defined with respect to prices $p_1, \ldots, p_n$ and allocation a. Let us assume now that $|b^{-1}(j)| < t$. Clearly, there can be no consumer $i \in C_j$ with $p_{b(i)} < p_j^*$, since allocation b is chosen optimally and there are available copies of product j left unsold. It follows that there must exist a consumer $i_0 \in L_j$ with $b(i_0) \neq j$ and $p_{b(i_0)} \geq p_j^*$. Under this assumption we will show that allocation b is not optimal. The following chain of conclusions follows solely from the optimality of a. Since $p_{a(i_0)} < p_{b(i_0)}$ it must be the case that product $b(i_0)$ is sold out under allocation a, i.e., $|a^{-1}(b(i_0))| = s(b(i_0))$. Then there must be some consumer i_1 with $b(i_1) \neq a(i_1) = b(i_0)$. For this consumer it must be true that either $p_{b(i_1)} \leq p_{a(i_0)}$ (including the case that $b(i_1) = \varnothing$) or product $b(i_1)$ is sold out under a. Otherwise, modifying a by setting $a(i_0) = b(i_0)$ and $a(i_1) = b(i_1)$ would result in a feasible allocation with strictly higher revenue. By repeatedly applying this argument we obtain a chain $i_0, i_1, \ldots, i_s$ of consumers with $b(i_k) = a(i_{k+1})$ and $p_{b(i_s)} \leq p_{a(i_0)}$ (or $b(i_s) = \varnothing$). We can assume that $b(i_k) \neq j$ for all k. To see this, note, that otherwise we could for every consumer $i_0 \in L_j$ with $b(i_0) \neq j$ find a distinct consumer i_k with $b(i_k) = j$, which would in turn imply that $|b^{-1}(j)| \geq t$. The above argument is also depicted in Figure 3.3. We can define a feasible allocation c by going backwards along the constructed chain of consumers and allocating to each consumer the product she received under allocation a except for consumer i_0, who will now receive product j. Formally, we let $c(i_k) = a(i_k)$ for $k = 1, \ldots, s$, $c(i_0) = j$ and $c(i) = b(i)$ for all remaining consumers. We observe that

$$\sum_{k=0}^{s} p_{c(i_k)} = p_j^* + \sum_{k=1}^{s} p_{c(i_k)} = p_j^* + \sum_{k=1}^{s} p_{a(i_k)} = p_j^* + \sum_{k=0}^{s-1} p_{b(i_k)} > \sum_{k=0}^{s} p_{b(i_k)},$$

where the last inequality follows from $p_{b(i_s)} \leq p_{a(i_0)} < p_j^*$, since $i_0 \in L_j$. This contradicts the optimality of allocation b and, thus, finishes the proof. □

In analogy to Theorem 3.3.1 we obtain the following bound on the price of anarchy.

Theorem 3.5.2. *The price of anarchy in the unit-demand max-buying pricing game is 2.*

Proof: Let strategies $P^{eq} = (P_1^{eq}, \ldots, P_n^{eq})$ define a Nash equilibrium. We want to lower bound the expected revenue of agent j. We define a (deterministic) strategy P_j^* for agent j by $\Pr(p_j = p_j^*) = 1$ and let $P^* = (P_{-j}^{eq}, P_j^*)$ denote the modified set of strategies. By the definition of Nash equilibria we have that

$$\mathrm{E}_{(P_{-j}^{eq}, P_j^*)}\left[\, R_j \,\right] \leq \mathrm{E}_{P^{eq}}\left[\, R_j \,\right].$$

By Lemma 3.5.1 we can lower bound the expected revenue of agent j playing strategy P_j^* by

$$\mathrm{E}_{(P_{-j}^{eq}, P_j^*)}\left[\, R_j \,\right] \geq \sum_{t=0}^{|C_j|} t \cdot p_j^* \cdot \Pr_{P^{eq}}\big(|L_j| = t\big).$$

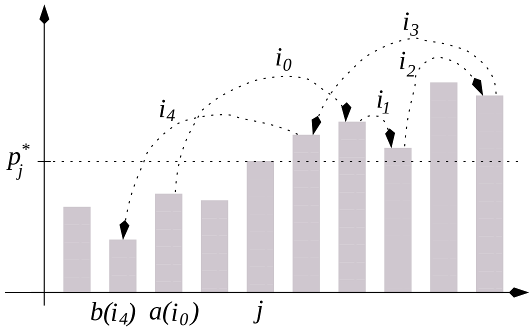


Figure 3.3: A chain of consumers switching to new products as constructed in the proof of Lemma 3.5.1, where $i_0 \in L_j$.

We can then write that

$$\begin{aligned}
\mathrm{E}_{P^{eq}}\big[\,R_j\,\big] + \mathrm{E}_{P^{eq}}\big[\sum_{i\in C_j} p_{a(i)}\,\big] &\geq \mathrm{E}_{(P^{eq}_{-j},P^*_j)}\big[\,R_j\,\big] + \mathrm{E}_{P^{eq}}\big[\sum_{i\in C_j} p_{a(i)}\,\big] \\
&\geq \sum_{t=0}^{|C_j|} t\cdot p_j^*\cdot \mathrm{Pr}_{P^{eq}}\big(|L_j| = t\big) + \sum_{t=0}^{|C_j|} t\cdot p_j^*\cdot \mathrm{Pr}_{P^{eq}}\big(|H_j| = t\big) \\
&= \sum_{t=0}^{|C_j|} \mathrm{Pr}_{P^{eq}}\big(|L_j| = t\big)\cdot p_j^*\cdot |C_j| = p_j^*\cdot |C_j|,
\end{aligned}$$

where we use the fact that

$$\mathrm{Pr}_{P^{eq}}\big(|H_j| = t\big) = \mathrm{Pr}_{P^{eq}}\big(|L_j| = |C_j| - t\big).$$

Let R denote the revenue of the equilibrium state, r^* the revenue generated by the optimal solution. By using linearity of expectation (see Appendix A.2) we have that

$$\begin{aligned}
2\cdot \mathrm{E}_{P^{eq}}\big[R\big] &= \sum_{j\in\mathcal{P}} \mathrm{E}_{P^{eq}}\big[\,R_j\,\big] + \mathrm{E}_{P^{eq}}\big[\sum_{i\in\mathcal{C}} p_{a(i)}\big] \\
&= \sum_{j\in\mathcal{P}}\Big(\mathrm{E}_{P^{eq}}\big[\,R_j\,\big] + \mathrm{E}_{P^{eq}}\big[\sum_{i\in C_j} p_{a(i)}\,\big]\Big) \geq \sum_{j\in\mathcal{P}} p_j^*\cdot |C_j| = r^*.
\end{aligned}$$

This gives the desired upper bound on the price of anarchy. We now give a simple corresponding lower bound. Consider a problem instance with 2 products $\mathcal{U} = \{u_1, u_2\}$ each of which is available only once, i.e., $s(u_1) = s(u_2) = 1$, and 2 consumers $\mathcal{C} = \{c_1, c_2\}$ with budgets $b(c_1, u_1) = \varepsilon$, $b(c_1, u_2) = 1$, $b(c_2, u_1) = 1$ and $b(c_2, u_2) = 1 + \varepsilon$. It is easy to see that the optimal solution generates revenue 2, while the pure strategies $p_1 = \varepsilon$ and $p_2 = 1 + \varepsilon$ define a Nash equilibrium which results in overall revenue $1 + 2\varepsilon$. Thus, the above bound is tight. $\square$

We point out that the situation is quite different in the min-buying or rank-buying models. For both models it is straightforward to show that the price of anarchy in the pricing game defined as above is unbounded, which is essentially due to the lack of the revenue transfer property (see Section 3.3).

3.6 Literature

The max-buying model has first been introduced in [Rus03]. The first theoretical results are found in [AFMZ04]. Here, Aggarwal et al. present an LP-based 1.59-approximation and a proof of APX-hardness for the general unlimited-supply case and derive the PTAS for the unlimited-supply price-ladder scenario. It is also shown that the dynamic programming approach can be extended to obtain a 4-approximation for the case of a price-ladder and limited supply. A better approximation guarantee for the combination of a price-ladder constraint and limited supply is not known. This is quite interesting, as one would expect the problem to be more difficult with the price-ladder removed, yet we have seen that local search (which is not applicable in the presence of a price-ladder constraint) yields a 2-approximation in this case.

Pricing with limited supply is also considered by Guruswami et al. [GHK$^+$05] in the context of unit-demand envy-free pricing (see Chapter 5). Our definition of strictly feasible allocations is inspired by their notion of envy-freeness.

A general introduction to non-cooperative games and Nash equilibria as a concept of rational player behavior is found, e.g., in [Owe95]. Investigation of the price of anarchy as a measure of the social cost of lack of coordination in distributed systems has been initiated by Koutsoupias and Papadimitriou in [KP99]. The new results in this chapter are found in [BK07].

4 The Space Between: Stochastic Selection and the Rank-Buying Model

Let us briefly recapitulate our results on unit-demand pricing up to this point. We have seen that the max-buying model allows us to design algorithms that achieve constant approximation guarantees. Yet, these results are not completely satisfactory in the sense that slight doubts may prevail as to what extend this model captures rational consumer behavior. On the other hand, the min-buying model is considered to be significantly more realistic, but has turned out to be intractable beyond the non-constant approximation obtained by the single-price algorithm, which unfortunately renders it rather irrelevant for practical purposes. Consequently, it is a very natural question to ask whether one can come up with economically realistic versions of unit-demand pricing that allow reasonable approximation ratios. In this chapter we will investigate two quite different approaches to this task.

Clearly, one could try to define a new selection rule that is sort of in-between max- and min-buying, in the sense that it is close enough to min-buying to capture rational consumer behavior, but also close enough to max-buying to be computationally tractable. We will show that this approach cannot be successful. To capture a wide range of selection rules that are based on product prices and are situated between the max- and min-buying models, we define the notion of *order-based stochastic selection rules*, which for each consumer define a probability distribution over the set of affordable products depending only on the relative order of prices, the problem's objective becoming maximization of the expected revenue from the resulting sales. We obtain a class of selection rules that model a wide range of consumer behavior, with max- and min-buying as the extremes at both ends of the chart. For the distribution-based problem version we prove that constant approximation ratios are possible for selection rules that mimic the max-buying model, while non-constant lower bounds hold for essentially every other order-based selection rule. Especially, even the case in which a consumer chooses one of her affordable products purely at random (the *random-buying* model) turns out to be no more tractable than min-buying itself. These results can be extended to the sampling-based problem version in a slightly weaker form.

An alternative approach lies in designing selection rules that do not depend on the relative order of product prices. The *rank-buying* model assumes that each consumer comes with a personal ranking of the products she is interested in and purchases the highest ranked affordable product. As it turns out, the rank-buying model is as intractable as min-buying in general. Yet, a small and natural restriction makes a lot of difference here. Assuming that a consumer's budgets are consistent with her ranking, i.e., higher ranked products are assigned non-smaller budgets, rank-buying reduces to max-buying in the presence of a price-ladder constraint and, thus, allows a polynomial time approximation scheme.

Section 4.1 contains formal definitions of stochastic selection rules and the rank-buying model. The results on pricing with stochastic selection rule are found in Section 4.2. The rank-buying model is investigated in Section 4.3. Section 4.4 points to some related literature.

4.1 Preliminaries

We define stochastic selection as a natural extension of general selection rules that were introduced in Section 2.1. A *stochastic selection rule* s is a function that, given fixed prices p, assigns to each pair of consumer $c \in \mathcal{C}$ and product $u \in \mathcal{U}$ the probability $s(c,u,p) \in [0,1]$ of consumer c buying u under price assignment p. We assume that a consumer never selects a product she cannot afford and always selects a product if she can afford to do so. Formally, we require that $s(c,u,p) = 0$ if $u \notin A_c(p)$, i.e., $p(u) > b(c,u)$, and that $\sum_{u \in A_c(p)} s(c,u,p) = 1$ if $A_c(p) \neq \emptyset$.

Definition 4.1.1. *Given products $\mathcal{U}$, consumer samples $\mathcal{C}$ consisting of budgets $b(c,u) \in \mathbb{R}_0^+$ for all $c \in \mathcal{C}$, $u \in \mathcal{U}$ and stochastic selection rule s,* UDP($\mathcal{C}$)-s *asks for a price assignment p maximizing the expected revenue*

$$r_s(p) = \sum_{c \in \mathcal{C}} \sum_{u \in \mathcal{U}} s(c,u,p) \cdot p(u).$$

In distribution-based UDP($\mathcal{D}$)-s *with (finite support) distribution $\mathcal{D}$ over consumer space $\mathcal{C}$ we want to maximize the expected revenue*

$$r_s(p) = \sum_{c \in \mathcal{C}} Pr_{\mathcal{D}}(c) \sum_{u \in \mathcal{U}} s(c,u,p) \cdot p(u)$$

from a sale to a single consumer drawn from $\mathcal{C}$ according to $\mathcal{D}$.

An alternative approach to model consumer behavior is to assume that products are not chosen according to the relative order of prices, but according to some consumer specific price-independent preferences. In the *rank-buying* model, each consumer c is represented by her budgets $b(c,u)$ for different products and a consumer-specific *ranking* $r_c : \mathcal{U} \to [|\mathcal{U}|]$, where $r_c(u) \neq r_c(u')$ for any $u \neq u'$.

Definition 4.1.2. *Given products $\mathcal{U}$, consumer samples $\mathcal{C}$ consisting of budgets $b(c,u) \in \mathbb{R}_0^+$ for all $c \in \mathcal{C}$, $u \in \mathcal{U}$ and rankings $r_c : \mathcal{U} \to [|\mathcal{U}|]$ as described above,* UDP($\mathcal{C}$)-RANK *asks for a price assignment p maximizing*

$$r_{rank}(p) = \sum_{c \in \mathcal{A}(p)} p\Big(\operatorname{argmin}\big\{r_c(u) \mid u \in A_c(p)\big\}\Big).$$

Given a price-ladder constraint π, UDP($\mathcal{C}$)-RANK-PL *asks for a revenue maximizing price assignment satisfying this constraint.*

4.2 Hardness of Stochastic Selection

General stochastic selection rules capture all possible kinds of selection. We are interested in a restricted class of selection rules, that select products according to their price relative to other affordable alternatives. Given some consumer $c \in \mathcal{C}$ and prices p, let $\pi_c^p : A_c(p) \to [|A_c(p)|]$ be a ranking such that $\pi_c^p(u) \leq \pi_c^p(u')$ iff $p(u) \geq p(u')$, i.e., π_c^p ranks products in $A_c(p)$ according to their non-increasing prices.

Definition 4.2.1. *Stochastic selection rule s is said to be* order-based *if $s(c,u,p) = s(\pi_c^p(u), |A_c(p)|)$, i.e., if the probability of c buying u depends only on the number of affordable products and the rank of product u among them.*

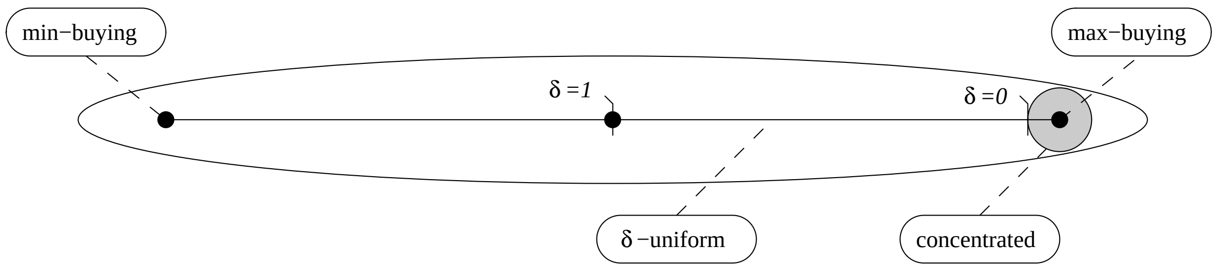


Figure 4.1: Order-based stochastic selection rules naturally fill the gap between the min- and max-buying models. The case $\delta = 0$ corresponds to selection rules concentrated around the max-buying end of the chart, $\delta = 1$ means that consumers choose their products uniformly at random.

The following definition classifies order-based selection rule based on their distance to the max-buying model. Fig. 4.1 illustrates the relation of these classes to the min- and max-buying models.

Definition 4.2.2. *Order-based stochastic selection rule s is called* concentrated, *if* $s(1, |A_c(p)|) = \Omega(1)$. *It is called* δ-uniform, *if* $s(1, |A_c(p)|) = \mathcal{O}(|A_c(p)|^{-\delta})$ *for* $0 \leq \delta \leq 1$.

Observe that by our definition above order-based selection rules need not be either concentrated or uniform. However, many natural selection rules fall into these classes. Intuitively, concentrated selection rules are very close to the max-buying model, since every consumer acts according to the max-buying model with some constant probability. On the other hand, in every selection rule modeling rational consumer behavior, the probability of selecting the most expensive product should decrease as more affordable alternatives become available. Thus, we would expect every realistic selection rule to be δ-uniform for some $\delta > 0$. It is a straightforward observation that the known constant factor approximation algorithms for the max-buying objective yield constant approximation guarantees when applied to UDP($\mathcal{C}$)-s or UDP($\mathcal{D}$)-s with concentrated selection rules as well (see, e.g., Theorems 3.3.2 and 3.3.3).

The interesting question is what can be said about the approximability of the problem with δ-uniform selection rules. We will focus first on distribution-based UDP($\mathcal{D}$)-s, for which it turns out that we can get tight inapproximability results for any value of δ. This shows that among all selection rules based on the relative order of product prices, constant approximation guarantees are possible only for those essentially mimicking the max-buying model. The proof of Theorem 4.2.3 is based on a randomized procedure that given arbitrary solutions to UDP($\mathcal{D}$)-s detects a selection of products that in expectation generates good profit under the min-buying objective, as well.

Theorem 4.2.3. *Let s be δ-uniform for some $0 < \delta < 1$. Then* UDP($\mathcal{D}$)-s *is not approximable within* $\mathcal{O}(n^{\delta-\varepsilon})$ *for any $\varepsilon > 0$, unless* P = NP. *On the other hand, if s is concentrated, i.e., $\delta = 0$, then* UDP($\mathcal{D}$)-*s allows constant approximation ratios.*

Proof: Let an instance of UDP($\mathcal{D}$)-MIN as in Corollary 2.3.11 be given. These instances are as defined in the proof of Theorem 2.3.7 with $\Delta = n - 1$ and consumers $C_{i,j}$ represented by a single consumer $c_{i,j}$ with probability $|\mathcal{C}_{i,j}|/|\mathcal{C}|$. More precisely, we are given products $\mathcal{U} = \{u_1, \dots, u_n\}$ and consumers $\mathcal{C} = \{c_1, \dots, c_n\}$, where each consumer c_j has budget value p_j for product u_j and budgets of at most

$\mu^{-1}p_j$ for any other product, where $\mu = 2n$. By Corollary 2.3.11 it is NP-hard to approximate UDP($\mathcal{D}$)-MIN on these instances within $\mathcal{O}(n^{1-\varepsilon})$ for any $\varepsilon > 0$.

Let s be δ-uniform stochastic selection rule. We proceed by introducing some notation that will be used throughout the remainder of the proof. Given a price assignment p and a consumer $c \in \mathcal{C}$ we let $r_{\min}(p \,|\, c)$ refer to the expected revenue made by sales at prices p to consumer c under the min-buying objective. Analogously, we denote the expected profit under selection rule s by $r_s(p \,|\, c)$. For subsets $\mathcal{C}' \subseteq \mathcal{C}$ of consumers we define $r_{\min}(p \,|\, \mathcal{C}') = \sum_{c \in \mathcal{C}'} r_{\min}(p \,|\, c)$ and $r_s(p \,|\, \mathcal{C}') = \sum_{c \in \mathcal{C}'} r_s(p \,|\, c)$. Optimal revenue under both objectives is denoted by r^*_{min} and r^*_s, respectively.

Assume now towards a contradiction that we can approximate UDP($\mathcal{D}$)-s within $\mathcal{O}(n^{\delta-\varepsilon})$ for some small $\varepsilon > 0$ and let p be the corresponding price assignment on the above instance. Thus, we have that

$$r_s(p \,|\, \mathcal{C}) \geq \frac{1}{\mathcal{O}(n^{\delta-\varepsilon})} r^*_s \geq \frac{1}{\mathcal{O}(n^{\delta-\varepsilon})} r^*_{\min},$$

where the last inequality follows from the fact that under s each consumer buys at a non-smaller price than under the min-objective with probability 1.

As in the proof of Theorem 2.3.7 one can argue that at least half the expected revenue $r_s(p \,|\, \mathcal{C})$ is contributed by consumers c_j who can afford to buy their associated product u_j, i.e., for which $p(u_j) \leq p_j$. Let us again refer to these consumers as $\mathcal{C}^+$. Let

$$A_j(p) = \{u_k \,|\, p(u_k) \leq b(c_j, u_k,)\}$$

refer to the set of products that consumer c_j can afford to buy under price assignment p. We partition the set of consumers into $\mathcal{C}^+ = \mathcal{C}^0 \cup \ldots \cup \mathcal{C}^\nu$, where

$$\mathcal{C}^q = \{c_j \,|\, 2^q \leq |A_j(p)| < 2^{q+1}\}.$$

Note, that clearly $\nu = \mathcal{O}(\log n)$, since the size of each $A_j(p)$ is trivially upper bounded by n. Fix $0 \leq r \leq \nu$ such that $r_s(p \,|\, \mathcal{C}^r) \geq r_s(p \,|\, \mathcal{C}^q)$ for all q and we obtain by the pigeon hole principle that

$$r_s(p \,|\, \mathcal{C}^r) \quad \geq \quad \frac{1}{\mathcal{O}(n^{\delta-\varepsilon} \log n)} r^*_{\min} \geq \frac{1}{\mathcal{O}(n^{\delta-\varepsilon+\rho})} r^*_{\min} \tag{4.1}$$

for any $\rho > 0$. As in the proof of Theorem 2.3.7 it is w.l.o.g. that $p(u_j) \in \{p_j, p_j + \mu\}$ for some small $\mu > 0$. We now define price assignment p' by the following random experiment that is repeated independently for each u_j. With probability 2^{-r-1} we set $p'(u_j) = p(u_j)$. With probability $1 - 2^{-r-1}$ we set $p'(u_j) = p_j + \mu$.

Consider consumer $c_j \in \mathcal{C}^r$. After the random experiment, this consumer buys product u_j at price p_j under the min-buying objective if its price remains unchanged and the prices of all other affordable products are set above their threshold prices. Thus, we can write that

$$\Pr\left(c_j \text{ buys } u_j \text{ at price } p_j \text{ under } p'\right) \geq \frac{1}{2^{r+1}} \left(1 - \frac{1}{2^{r+1}}\right)^{|A_j(p)|-1} \geq \frac{1}{\mathrm{e} \cdot 2^{r+1}},$$

where we use that $|A_j(p)| < 2^{r+1}$ by the fact that $c_j \in \mathcal{C}^r$, and obtain an expected contribution of

$$\mathrm{E}\left[r_{\min}(p' \,|\, c_j)\right] \geq \frac{1}{\mathrm{e} \cdot 2^{r+1}} \cdot p_j$$

by consumer c_j under price assignment p'. It remains to show an upper bound on the expected revenue made by c_j with selection rule s under price assignment p, which follows easily from the δ-uniformity of s. More precisely, we know that consumer c_j buys u_j at price p_j with probability at most $|A_j(p)|^{-\delta} < 2^{-\delta r}$. Other products can be bought at price at most $\mu^{-1} p_j$. Now remember that $\mu = \Omega(2^r)$ and we obtain

$$r_s(p \,|\, c_j) \leq 2^{-\delta r} p_j + \mu^{-1} p_j = \mathcal{O}(\left(\frac{1}{2^r}\right)^{\delta}) \cdot p_j,$$

and, using that $2^r \leq n$,

$$\mathrm{E}\left[r_{\min}(p' \,|\, c_j)\right] \quad = \quad \frac{1}{\mathcal{O}(n^{1-\delta})} r_s(p \,|\, c_j). \tag{4.2}$$

Summing over all $c_j \in \mathcal{C}^r$ this yields that

$$\begin{aligned} \mathrm{E}\left[r_{\min}(p' \,|\, \mathcal{C}^r)\right] &= \frac{1}{\mathcal{O}(n^{1-\delta})} r_s(p \,|\, \mathcal{C}^r), && \text{by (4.2)} \\ &= \frac{1}{\mathcal{O}(n^{1-\varepsilon+\rho})} r^*_{\min}, && \text{by (4.1).} \end{aligned}$$

By choosing ρ small enough, we obtain a solution with approximation guarantee $\mathcal{O}(n^{1-\varepsilon'})$ for some fixed $\varepsilon' > 0$ for UDP($\mathcal{D}$)-MIN. Thus, we have shown that UDP($\mathcal{D}$)-s with δ-uniform selection rule does not allow approximation guarantees essentially better than n^{δ}, which finishes the first part of the proof.

We have seen before that UDP($\mathcal{D}$)-MAX can be approximated within constant ratios. It is then straightforward to argue that going from the max-buying objective to a concentrated stochastic selection rule, the overall revenue is reduced by at most a constant factor, since every consumer is still going to buy the most expensive affordable product with constant probability. □

Let us briefly point out in what sense the inapproximability results for UDP($\mathcal{D}$)-s with δ-uniform s in Theorem 4.2.3 are tight. The notion of δ-uniformity requires that $s(1, |A_c(p)|) = \mathcal{O}(|A_c(p)|^{-\delta})$. Especially, every γ-uniform selection rule with $\gamma > \delta$ is also δ-uniform. This makes it impossible to get tight approximation results for UDP($\mathcal{D}$)-s with δ-uniform selection rules. However, if we require that $s(1, |A_c(p)|) = \Theta(|A_c(p)|^{-\delta})$ instead, we immediately obtain a matching $\mathcal{O}(n^{\delta})$-approximation by a reduction to the max-buying model.

So far, we have only considered distribution-based UDP($\mathcal{D}$)-s. Clearly, the positive results for concentrated selection rules trivially carry over to sampling-based UDP($\mathcal{C}$)-s. But what about the lower bounds for the δ-uniform case? Here we can get similar results only for a certain range of δ-values. This is due to the fact that our lower bounds for the min-buying model are not perfectly tight. Remember that the single-price algorithm achieves approximation ratio H_m for UDP($\mathcal{C}$)-MIN, but our lower bound is only $\Omega(\log^{\varepsilon} m)$ for some $\varepsilon > 0$. In this notation, Theorem 4.2.4 holds for any $\delta > 1 - \varepsilon$. It is quite evident that the gap in Theorem 4.2.4 is an artifact of its current proof and, in fact, Theorem 4.2.3 is likely to hold for sampling-based UDP($\mathcal{C}$)-s, as well.

Theorem 4.2.4. *Let s be δ-uniform for some sufficiently large $0 < \delta < 1$. Then* UDP($\mathcal{C}$)-s *is not approximable within $\mathcal{O}(\log^{\varepsilon} m)$ for some $\varepsilon > 0$, unless* NP $\subseteq$ DTIME($n^{O(\log \log n)}$).

4.3 Approximability of Rank-Buying

We next turn to the rank-buying model. As we shall see, several of the results for the min- and max-buying models can be more or less directly applied to the rank-buying model, giving an almost complete characterization of this model's complexity.

Given a price-ladder constraint, it is straightforward to encode any UDP($\mathcal{C}$)-MIN-PL or UDP($\mathcal{D}$)-MIN-PL instance in terms of the rank-buying model. In fact, we just need to define every consumer's ranking according to the price-ladder, i.e., $r_c(u) < r_c(u')$ whenever $p(u) \leq p(u')$ is required by the price-ladder. Using these rankings, every consumer is going to buy the cheapest product she can afford under any given price assignment. Hence, all hardness results for UDP($\mathcal{C}$)-MIN-PL and UDP($\mathcal{D}$)-MIN-PL carry over to UDP($\mathcal{C}$)-RANK-PL and UDP($\mathcal{D}$)-RANK-PL. It is also straightforward to argue that the proof of Theorem 2.3.7 works for rank-buying without price-ladder, as well, since in the resulting pricing instance the relative order of prices remains fixed even without an explicit price-ladder constraint. Consequently, the same argumentation as before applies and we obtain similar hardness results for UDP($\mathcal{C}$)-RANK and UDP($\mathcal{D}$)-RANK. Theorem 4.3.1 is analogous to Corollaries 2.3.8 through 2.3.10.

Theorem 4.3.1. UDP($\mathcal{C}$)-RANK *and* UDP($\mathcal{C}$)-RANK-PL *are not approximable within* $\mathcal{O}(\log^{\varepsilon} m)$ *for some* $\varepsilon > 0$*, unless* NP $\subseteq$ DTIME($n^{\mathcal{O}(\log \log n)}$). *Allowing at most* $\ell \geq 3$ *non-zero budgets per consumer it is not approximable within* ℓ^{ε} *for some* $\varepsilon > 0$*, unless* P = NP. *Furthermore, it is not approximable within* $\mathcal{O}(n^{\varepsilon})$ *for some* $\varepsilon > 0$*, unless* NP $\subseteq \bigcap_{\delta>0}$ DTIME($2^{\mathcal{O}(n^{\delta})}$).

In analogy to Corollary 2.3.11 we get inapproximability of the distribution-based problem version.

Theorem 4.3.2. UDP($\mathcal{D}$)-RANK *and* UDP($\mathcal{D}$)-RANK-PL *are not approximable within* $\mathcal{O}(n^{1-\varepsilon})$ *for any* $\varepsilon > 0$*, unless* P = NP.

On the algorithmic side, we have already seen in Section 2.2 that the single-price algorithm can be applied to all versions of unit-demand pricing, including the rank-buying model. The random-partitioning algorithm can be applied in the rank-buying scenario by the simple observation that the analysis in the proof of Theorem 2.4.1 counts revenue only from those consumers that can afford exactly one product. Thus, the selection rule in place is completely irrelevant.

Theorem 4.3.3 ([BB06]). *The random-partitioning algorithm (Algorithm 2) computes an (expected)* $\mathcal{O}(\ell)$*-approximation for* UDP($\mathcal{C}$)-RANK *with at most* ℓ *non-zero budgets per consumer.*

In [AFMZ04] a restricted version of the rank-buying model, in which a consumer's budget values need to be consistent with her ranking, has been considered. More formally, UDP($\mathcal{C}$)-RANK *with consistent budgets* requires that for every consumer $c \in \mathcal{C}$, we have that $b(c, u) \geq b(c, u')$ whenever $r_c(u) < r_c(u')$ for all products $u, u' \in \mathcal{U}$. Given a price-ladder constraint, UDP($\mathcal{C}$)-RANK-PL reduces to the max-buying model.

Theorem 4.3.4 ([AFMZ04]). UDP($\mathcal{C}$)-RANK-PL *with consistent budgets reduces to* UDP($\mathcal{C}$)-MAX-PL.

Proof: Let an instance of UDP($\mathcal{C}$)-RANK-PL be given and fix some consumer c. We show that we can transform the instance into an equivalent instance in which consumer c buys the most expensive affordable product for any price assignment p.

Fix prices p, such that c can afford products u and u' with prices $p(u) < p(u')$ and assume that c selects u. Note, that by the price-ladder constraint $p(u) \leq p(u')$ must be true for any price assignment. Since c chooses u, we have that $r_c(u) < r_c(u')$. By the fact that budgets are consistent it follows that $b(c, u) \geq b(c, u')$. Consequently, whenever c can afford to buy u' the same is true for u and, thus, c will never buy product u'. Thus, we obtain an equivalent UDP($\mathcal{C}$)-RANK-PL instance by setting $b(c, u') = 0$ and assigning the lowest possible rank $r_c(u') = n - 1$ to u'. (Here the ranks of other products need to be adapted accordingly.)

The above step decreases the number of on-zero budgets by 1 and, thus, we eventually arrive at an UDP($\mathcal{C}$)-RANK-PL instance in which each consumer always chooses to purchase the most expensive affordable product. □

By Theorem 4.3.4 the PTAS from Section 3.4.1 can be applied to the rank-buying model. It is also straightforward to modify the proof of Theorem 3.4.2 in order to fit rank-buying. We thus obtain a similar matching lower bound.

Theorem 4.3.5 ([AFMZ04]). UDP($\mathcal{C}$)-RANK-PL *with consistent budgets allows a PTAS.*

Theorem 4.3.6. UDP($\mathcal{C}$)-RANK-PL *with consistent budgets is strongly* NP-*hard, even if each consumer has at most* 2 *non-zero budgets.*

4.4 Literature

The notion of stochastic selection rules was introduced in [Bri06], where also the results from Section 4.2 are found.

The rank-buying model was first considered by Rusmevichientong in [Rus03]. The notion of consistent budgets is implicit in [AFMZ04] where Aggarwal et al. prove that rank-buying with consistent budgets reduces to max-buying in the presence of a price-ladder and derive the PTAS for this problem.

The new results from Section 4.3 have been published in [BK07].

5 Uniform Budgets: The Envy-Free Pricing Problem

In the previous chapter we have seen that neither stochastic selection rules nor deterministic selection rules based on external rankings result in algorithmically tractable variations of unit-demand pricing in the no-price-ladder scenario. We will now investigate a different approach towards this goal, which is based on remaining true to the min-buying model as our most basic model of rational behavior, yet restricting the problem in another way.

The negative results on general min-buying in Chapter 2 heavily rely on the fact that consumers can express preferences over individual products by assigning different budget values. This allowed us to bind groups of consumers to specific products and encode independence between them. A natural problem restriction that circumvents this kind of hardness is the *uniform-budget case*, in which we do not allow consumers to distinguish between products they are interested in, in the sense that their budgets for all desired products must be identical.

This problem variation is also interesting for another reason. In economic or game-theoretic settings a player's happiness is often measured in terms of her *utility*, which is commonly defined as the difference between the player's valuation for the game's outcome and the payment she is charged for being allowed to take part in the game. In the setting of unit-demand pricing, we can view a consumer's utility as the difference between her budget for the product she buys and its actual price. This model of consumer behavior, in which consumers choose the product maximizing their utility, was first proposed in [AFMZ04] and is usually referred to as *max-gain buying* in the unlimited-supply setting.

The game-theoretic flavor of this problem variation becomes even more evident in the limited-supply scenario, when it is not guaranteed that under a given pricing different consumers can simultaneously receive their preferred products. In [GHK$^+$05], Guruswami et al. propose the *unit-demand envy-free pricing problem*. Given a limited supply of products, we need to assign revenue maximizing prices, such that every consumer can purchase the product maximizing her utility. The additional requirement that every consumer must receive her most desired product is termed *envy-freeness*. Note, that envy-freeness is obviously not an issue in unlimited-supply settings.

In [AFMZ04] it is shown that the single-price algorithm achieves the same approximation guarantees for max-gain buying as for all other models. In [GHK$^+$05] these results are extended to limited product-supply. Here, previous work in economics guarantees the existence of envy-free pricings for unit-demand utility functions and extending these by reserve prices obtained from the single-price algorithm yields the approximation guarantee.

Does unit-demand envy-free pricing allow better approximation guarantees than those obtained by the extended single-price algorithm? Clearly, unlimited-supply max-gain buying is a special case of the envy-free pricing problem. Furthermore, restricting the problem even more and assuming uniform budgets, max-gain buying coincides with the min-buying model, since in this case it is always the cheapest affordable product that maximizes a consumer's utility. Consequently, any lower bound on the approximability of uniform-budget min-buying yields the same result for unit-demand envy-free pricing.

We will show here that uniform-budget min-buying is unlikely to be approximable within any better than the known ratios. More formally, we will derive inapproximability under an assumption about the average case complexity of refuting random 3SAT-instances.

The rest of this chapter is organized as follows. Section 5.1 contains the formal problem definitions. Section 5.2 contains a high-level description of the result. The formal proofs are contained in Section 5.3. Section 5.4 gives an overview of related literature.

5.1 Preliminaries

As described before, we will consider the restriction of UDP($\mathcal{C}$)-MIN in which each consumer has only a single non-zero budget for a number of products she is interested in. Formally, for every consumer c there exist $S_c \subseteq \mathcal{U}$ and $b_c \in \mathbb{R}^+$, such that $b(c,u) = b_c$ for all $u \in S_c$, $b(c,u) = 0$ else. For the sake of completeness we give a formal definition below.

Definition 5.1.1. *In* uniform-budget UDP($\mathcal{C}$)-MIN *we are given products $\mathcal{U}$ and consumer samples $\mathcal{C}$ consisting of budgets $b_c \in \mathbb{R}^+$ and product sets $S_c \subseteq \mathcal{U}$ for all $c \in \mathcal{C}$. We want to find prices $p : \mathcal{U} \to \mathbb{R}_0^+$ that maximize*

$$r_{\min}(p) = \sum_{c \in \mathcal{A}(p)} \min\{p(u) \mid u \in S_c \wedge p(u) \leq b_c\},$$

where again $\mathcal{A}(p)$ denotes the set of consumers that can afford any product under p.

Additionally assuming limited product supply and requiring an envy-free allocation of the products, we get the unit-demand envy-free pricing problem, which is defined next.

Definition 5.1.2. *Given products $\mathcal{U}$, each $u \in \mathcal{U}$ available in supply $s(u) \in \mathbb{N}$, and consumer samples $\mathcal{C}$ consisting of budgets $b(c,u)$, the* unit-demand envy-free pricing problem *asks for prices $p : \mathcal{U} \to \mathbb{R}_0^+$ and allocation $a : \mathcal{C} \to \mathcal{U}$, such that*

$$a(c) = \operatorname{argmax}\big\{b(c,u) - p(u) \mid u \in \mathcal{U} \cup \{\varnothing\}\big\},$$

$|a^{-1}(u)| \leq s(u)$ for all $u \in \mathcal{U}$, and the overall revenue $\sum_{c \in \mathcal{C}} p(a(c))$ is maximized, where by definition $p(\varnothing) = b(c,\varnothing) = 0$ for all $c \in \mathcal{C}$.

Observe that uniform-budget UDP($\mathcal{C}$)-MIN is a special case of both max-gain buying and the unit-demand envy-free pricing problem, since with uniform budgets it is always the product with lowest absolute price that maximizes a consumer's utility. Distribution-based UDP($\mathcal{D}$)-MIN with uniform budgets and the distribution-based version of envy-free pricing are defined analogously.

5.2 Hardness of Approximation - Overview

Our hardness proof for uniform-budget UDP($\mathcal{C}$)-MIN works along the lines of Theorem 2.3.7 in Chapter 2. The main similarity is the fact that we will once more resort to the idea of scaling the hardness of the base problem of our reduction to a level that yields inapproximability thresholds of the right magnitude while still being sparse enough to be encoded in terms of sampling-based unit-demand pricing. Instead of the independent set problem we will use the *Balanced Bipartite Independent Set Problem* (BBIS), which is defined below, as the starting point of our reduction. We give a relatively detailed overview of the reduction and its implications for uniform-budget UDP($\mathcal{C}$)-MIN below. The formal proof of the main result in Theorem 5.2.10 is found in Section 5.3.

Essentially, we show a reduction from BBIS in constant degree bipartite graphs to UDP($\mathcal{C}$)-MIN with uniform budgets. This shows that, assuming there are no randomized polynomial time algorithms of a certain kind approximating constant degree BBIS within arbitrarily small constant factors, there are no polynomial time algorithms approximating uniform-budget UDP($\mathcal{C}$)-MIN within $\mathcal{O}(\log^{\varepsilon} m)$ for some $\varepsilon > 0$.

To date, no explicit hardness results have been proven for BBIS in constant degree graphs, although the problem has been receiving a lot of attention. In [Fei02], Feige shows an interesting connection between the average case complexity of refuting 3CNF-formulas and the worst case approximation complexity of several notorious optimization problems including BBIS. To embed our result into a somewhat wider context, we formulate a slightly stronger version of the hypothesis in [Fei02] and show that this is enough for our purposes.

Remember that a 3CNF-formula is a conjunction of clauses, each of which is the disjunction of 3 literals over variables $x_1, \ldots, x_n$, where a literal is a variable or its negation. Before stating the hypothesis we need to describe the random sampling procedure used to obtain random 3CNF formulas in [Fei02]. Given n variables we create formulas consisting of $m = \Delta n$ clauses for some large constant $\Delta \in \mathbb{N}$. Each literal of every clause is picked uniformly at random from the set of $2n$ literals. Thus, every clause consists of 3 (not necessarily distinct) literals that are picked independently at random. When Δ is large enough, every truth assignment satisfies roughly $(7/8)m$ clauses of a random 3CNF formula. Thus, a *typical* random 3CNF formula does not have significantly more than $(7/8)m$ simultaneously satisfiable clauses. On the other hand, for a sufficiently small $\varepsilon > 0$ formulas with $(1 - \varepsilon)m$ simultaneously satisfiable clauses can be considered *exceptional*. In fact, this is even true for any fixed $\varepsilon < 1/8$, since the number of satisfiable clauses is sharply concentrated around its expectation. By choosing Δ sufficiently large, deviations by more than a small constant factor can be excluded with overwhelming probability. Hypothesis 5.2.1 states that it is hard to detect exceptional formulas on average.

Hypothesis 5.2.1. *For every fixed $\varepsilon > 0$ and sufficiently large constant $\Delta \in \mathbb{N}$, there is no (randomized) algorithm that runs in time $\mathcal{O}(t(n))$ and, given a random 3CNF formula with n variables and $m = \Delta n$ clauses, outputs* typical *with probability at least $1/2$, but outputs* exceptional *on every formula with $(1 - \varepsilon)m$ simultaneously satisfiable clauses with probability at least $1 - 1/2^{poly(n)}$.*

Choosing $t(n) = \text{poly}(n)$ the only difference between Hypothesis 5.2.1 and the hypothesis in [Fei02] is that we allow randomized algorithms that have exponentially small error probability when it comes to detecting exceptional formulas. We need this stronger version as a result of our reduction from BBIS to uniform-budget UDP($\mathcal{C}$)-MIN, which is partially based on a random construction that introduces an exponentially small one-sided error probability for detecting large independent sets. We are mostly interested here in the case of $t(n) = \text{poly}(n)$. However, similar to what we have seen in Chapter 2, going to other subexponential time bounds will allow us to obtain lower bounds for differently parametrized approximation goals. In analogy to [Fei02] we define a notion of hardness based on Hypothesis 5.2.1. We use slightly different notation compared to [Fei02] to reflect the difference in the underlying hypotheses.

Definition 5.2.2. *A problem is said to be R3SAT*$^\star(t(n))$*-hard, if having a (randomized) polynomial time algorithm (with exponentially small failure probability) for it refutes Hypothesis 5.2.1.*

Most importantly, R3SAT$^\star(t(n))$-hard problems do not allow polynomial time algorithms if we believe that Hypothesis 5.2.1 is true for the given choice of $t(n)$. As a byproduct of the fact that Hypothesis 5.2.1 also excludes certain randomized algorithms, R3SAT$^\star(t(n))$-hardness rules out this type of algorithm, too. We continue by giving a formal definition of BBIS, the base problem of our reduction.

Definition 5.2.3. *In the* Balanced Bipartite Independent Set Problem *(*BBIS*) we are given a bipartite graph* $G = (V, W, E)$*. We want to find maximum cardinality subsets of vertices* $V' \subset V$*,* $W' \subset W$ *with* $|V'| = |W'|$*, such that* $\{v, w\} \notin E$ *for all* $v \in V'$*,* $w \in W'$*.*

The first step in our proof of hardness for uniform-budget UDP($\mathcal{C}$)-MIN is mostly identical to proofs given in [Fei02], where hardness of general BBIS is derived. We do a slightly more careful analysis and obtain R3SAT*(poly(n))-hardness of BBIS in constant degree graphs. The proof is found in Section 5.3.1.

We point out that this part of the proof can be replaced by Hypothesis 5.2.4, which states that the gap variant of BBIS in constant degree graphs does not have randomized polynomial time algorithms with one-sided error (i.e., the decision variant does not belong to the complexity class RP). More formally, let $\mathcal{G}(a, d)$, $\mathcal{G}(b, d)$ be two families of bipartite graphs on $2n$ vertices with constant degree $d \in \mathbb{N}$ and maximum BBIS of size at most an or at least bn, respectively. Given $0 < a < b < 1$ and $d \in \mathbb{N}$ the problem BBIS(a, b, d) requires deciding whether $G \in \mathcal{G}(a, d)$ or $G \in \mathcal{G}(b, d)$ for a given graph $G \in \mathcal{G}(a, d) \cup \mathcal{G}(b, d)$. For our purposes Hypothesis 5.2.4 is fully sufficient.

Hypothesis 5.2.4. *There exist constants* $0 < a < b < 1$ *and* $d \in \mathbb{N}$*, such that* BBIS$(a, b, d) \notin$ RP.

Without expressing too much of an opinion about the validity of Hypothesis 5.2.4, it should be noted that it is certainly in accordance with our current knowledge and backed by the fact that strong super-constant approximability thresholds have been proven for general BBIS. Having hardness of constant degree BBIS we once more apply the method of derandomized graph products [AFWZ95] to obtain hardness of approximation within $\mathcal{O}(f(n)^{\varepsilon})$ for BBIS in graphs with maximum degree $\mathcal{O}(f(n))$. The application of this technique to balanced bipartite sets is sketched in Section 5.3.2. The formal proof of Theorem 5.2.5 is equivalent to the proof given in Section 2.3.1 for the case of regular independent sets.

Theorem 5.2.5. *Let* $f : \mathbb{N} \longrightarrow \mathbb{R}_+$ *be non-decreasing with* $f(n) \leq n$ *and* $f(n^c) \leq f(n)^c$ *for all* $c \geq 1$*,* $n \in \mathbb{N}$*. Let* $\mathcal{G}(a(n), f(n))$ *and* $\mathcal{G}(b(n), f(n))$ *be the families of balanced bipartite graphs on* $2n$ *vertices, with maximum degree bounded by* $f(n)$ *and maximum* BBIS *of size at most* $a(n)n$ *or at least* $b(n)n$*, respectively. There exist* $0 < a(n) < b(n) < 1$ *with* $b(n)/a(n) = \Omega(f(n)^{\varepsilon})$ *for some* $\varepsilon > 0$*, such that given* $G \in \mathcal{G}(a(n), f(n)) \cup \mathcal{G}(b(n), f(n))$ *it is R3SAT***(poly*(n)*)-hard to decide whether* $G \in \mathcal{G}(a(n), f(n))$ *or* $G \in \mathcal{G}(b(n), f(n))$*.*

We proceed by sketching the reduction of BBIS to uniform-budget UDP($\mathcal{C}$)-MIN. As an intermediate step we modify the BBIS instance by adding a number of random edges and interpret vertices on one side of the bipartition as sets. The connection to uniform-budget UDP($\mathcal{C}$)-MIN is made by considering sequences of these sets that have a certain expansion property, which is formalized in the following definition.

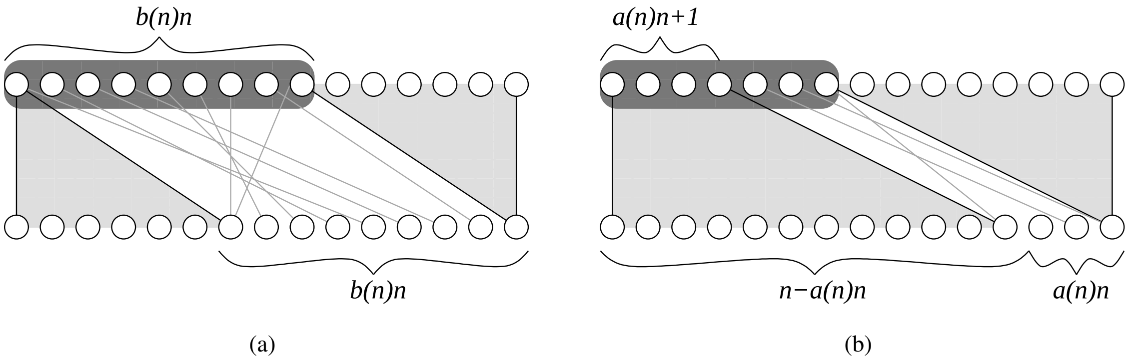


Figure 5.1: Reducing BBIS to MES. Let $0 < a(n) < b(n) < 1$ be defined as in Theorem 5.2.5. The top vertices are interpreted as sets containing their adjacent vertices from the opposite side of the bipartition. (a) If the graph has a balanced bipartite independent set of size at least $b(n)n$, adding random edges with probability $1/b(n)n$ each implants an expanding sequence of expected size $\Omega(b(n)n)$. (b) If the maximum balanced bipartite independent set has size $a(n)n$, every selection of $a(n)n + 1$ sets covers at least $n - a(n)n$ elements of the universe. Since every further set of an expanding sequence must cover an additional element, the length of the maximum expanding sequence is bounded above by $2a(n)n + 1$.

Definition 5.2.6. *In the* Maximum Expanding Sequence Problem *(*MES*) we are given an ordered collection* $S_1, \ldots, S_m$ *of sets. An expanding sequence* $\phi = (\phi(1) < \cdots < \phi(\ell))$ *of length* $|\phi| = \ell$ *is a selection of sets* $S_{\phi(1)}, \ldots, S_{\phi(\ell)}$*, such that*

$$S_{\phi(j)} \nsubseteq \bigcup_{i=1}^{j-1} S_{\phi(i)}$$

for $2 \leq j \leq \ell$*.* MES *asks for finding such a sequence of maximum length.*

We are not aware that MES has been considered explicitly before. Reducing general BBIS, for which inapproximability results under standard complexity theoretic assumptions have recently been proven by Khot [Kho04], yields strong hardness results for MES. The main idea of the proof of Theorem 5.2.7 is illustrated in Figure 5.1.

Theorem 5.2.7. MES *is inapproximable within* $\mathcal{O}(m^{\varepsilon})$ *for some* $\varepsilon > 0$*, unless* NP $\subseteq \bigcap_{\delta>0}$ BPTIME($2^{\mathcal{O}(n^{\delta})}$).

In order to reduce MES to uniform-budget UDP($\mathcal{C}$)-MIN we have to focus our attention on severely restricted problem instances. BBIS instances with maximum degree $f(n)$ yield MES instances that exhibit a nicely sparse structure. Definition 5.2.8 formalizes our notion of *sparse*.

Definition 5.2.8. *We say that an* MES *instance* $S_1, \ldots, S_m$ *is* $f(m)$-separable *if it can be partitioned into* $\kappa = \mathcal{O}(f(m))$ *subsequences* $\mathcal{C}_1, \ldots, \mathcal{C}_\kappa$*, such that* $\mathcal{C}_j = \{S_{k(j)}, S_{k(j)+1}, \ldots, S_{\ell(j)}\}$*, where* $k(1) = 1$*,* $\ell(\kappa) = m$*,* $k(j+1) = \ell(j) + 1$ *for* $1 \leq j \leq \kappa - 1$ *and each* $\mathcal{C}_j$ *contains only non-intersecting sets.*

It is actually not difficult to argue that BBIS instances with maximum degree $f(n)$ yield $f(m)^2$-separable MES instances. The degree bound implies that both the size of each set and the frequency of each element

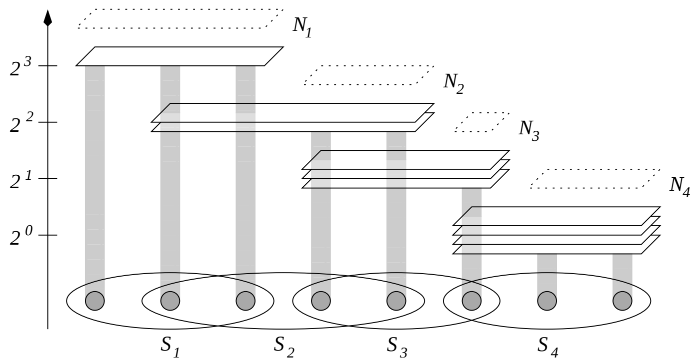


Figure 5.2: Sets S_1, S_2, S_3, S_4 form an expanding sequence. Each set S_i is transformed into a corresponding selection of consumers on price level 2^{4-i}. Defining N_i as the set of elements newly covered by set S_i, maximum profit from all consumers can be extracted by setting the prices of all elements in N_i to 2^{4-i}.

is bounded by $f(n)$. Thus, rearranging the order of the sets appropriately separates the instance into $f(n)^2$ (or $f(m)^2$, n and m being exchangeable in this context) blocks of non-intersecting sets. The proof of Lemma 5.2.9 is found in Section 5.3.3.

Lemma 5.2.9. *There exists $\varepsilon > 0$, such that* MES *with $f(m)$-separable instances is R3SAT*(poly(n))-hard to approximate within $\mathcal{O}(f(m)^\varepsilon)$.*

It is relatively straightforward to encode MES in terms of uniform-budget UDP($\mathcal{C}$)-MIN, since MES nicely models the dependence between different price levels in the pricing problem. Sets are transformed into corresponding consumers with exponentially decreasing budgets for the elements contained in the set. For consumers whose corresponding sets form an expanding sequence we can then find prices that ensure that they all buy at their budget values. This is depicted in Figure 5.2. The only difficulty lies in ensuring that the resulting uniform-budget UDP($\mathcal{C}$)-MIN instances are of polynomial size. It turns out that the notion of $f(m)$-separability is the key to this problem, since the non-intersecting sets belonging to a single block in the MES instance can be encoded on a single price level. Choosing $f(m) = \log m$ yields instances with logarithmically many price levels and, thus, a polynomial number of consumer samples. The formal proof of Theorem 5.2.10 is found in Section 5.3.4.

Theorem 5.2.10. *There exists $\varepsilon > 0$, such that it is R3SAT*(poly(n))-hard to approximate uniform-budget* UDP($\mathcal{C}$)-MIN *within $\mathcal{O}(\log^\varepsilon m)$. Hardness of approximation holds even under the weaker assumption of Hypothesis 5.2.4.*

Similar to what we have already seen in Chapter 2, the reduction is flexible enough to yield inapproximability results also in the maximum number ℓ of non-zero budgets per consumer and, allowing UDP($\mathcal{C}$)-MIN instances of subexponential size, we can stretch the construction to the limit and obtain lower bounds in terms of the number of products n, as well.

Theorem 5.2.11. *There exist constants $\ell_0 \in \mathbb{N}$ and $\varepsilon > 0$, such that for every $\ell \geq \ell_0$ it is R3SAT*(poly(n))-hard to approximate uniform-budget* UDP($\mathcal{C}$)-MIN *with at most ℓ non-zero budgets per consumer within*

ℓ^ε. Furthermore, for every $\delta > 0$ there exists $\varepsilon > 0$, such that it is R3SAT**$(2^{\mathcal{O}(n^\delta)})$-hard to approximate uniform-budget* UDP($\mathcal{C}$)-MIN *within $\mathcal{O}(n^\varepsilon)$. Hardness of approximation holds even under the weaker assumption of Hypothesis 5.2.4.*

For the sake of completeness we restate the results for the more general unit-demand envy-free pricing problem.

Corollary 5.2.12. *There exists $\varepsilon > 0$, such that it is R3SAT*(poly(n))-hard to approximate the unit-demand envy-free pricing problem within $\mathcal{O}(\log^\varepsilon m)$. There exist constants $\ell_0 \in \mathbb{N}$ and $\varepsilon > 0$, such that for every $\ell \geq \ell_0$ it is R3SAT*(poly(n))-hard to approximate the unit-demand envy-free pricing problem with at most ℓ non-zero budgets per consumer within ℓ^ε. For every $\delta > 0$ there exists $\varepsilon > 0$, such that it is R3SAT*$(2^{\mathcal{O}(n^\delta)})$-hard to approximate the unit-demand envy-free pricing problem within $\mathcal{O}(n^\varepsilon)$. Hardness of approximation holds even under the weaker assumption of Hypothesis 5.2.4.*

Finally, let us consider the distribution-based versions of uniform-budget UDP($\mathcal{D}$)-MIN and the unit-demand envy-free pricing problem. As mentioned before, a reduction similar to the one given in the proof of Lemma 5.2.9 in combination with the known hardness results for general BBIS from [Kho04] yields a strong hardness result for general MES. Applying the reduction from the proof of Theorem 5.2.10 we obtain inapproximability results for distribution-based uniform-budget UDP($\mathcal{D}$)-MIN under standard complexity theoretic assumptions. These immediately extend to the more general unit-demand envy-free pricing problem.

Theorem 5.2.13. UDP($\mathcal{D}$)-MIN *with uniform budgets is hard to approximate within $\mathcal{O}(n^\varepsilon)$ for some $\varepsilon > 0$, unless* NP $\subseteq \bigcap_{\delta>0}$ BPTIME$(2^{\mathcal{O}(n^\delta)})$. *The same hardness holds for the distribution-based unit-demand envy-free pricing problem.*

5.3 Full Proof of Theorem 5.2.10

The proof of Theorem 5.2.10 is organized as follows. Section 5.3.1 proves hardness of constant-degree BBIS based on Hypothesis 5.2.1. Hardness amplification via derandomized graph products is described in Section 5.3.2. The reduction from BBIS to MES is found in Section 5.3.3. Finally, Section 5.3.4 contains the reduction from MES to UDP($\mathcal{C}$)-MIN.

5.3.1 R3SAT*(poly(n))-hardness of Constant Degree BBIS

We show a reduction from MAX-3AND. Given a collection of clauses, each of which contains 3 (not necessarily distinct) literals and is satisfied if all 3 literals are assigned the boolean value true, we want to determine the maximum number of simultaneously satisfiable clauses. The remainder of this part of the proof is roughly identical to the one in [Fei02], except for the fact that a small change in the reduction yields graphs of constant degree. Lemma 5.3.1 is explicitly stated in [Fei02] for the case of their underlying hypothesis and extends easily to our notion of R3SAT*(poly(n))-hardness. We note that if we talk about random MAX-3AND instances, we assume the sampling procedure as described in Section 5.2.

Lemma 5.3.1 ([Fei02]). *For every fixed $\varepsilon > 0$ and sufficiently large constant $\Delta \in \mathbb{N}$, the following problem is R3SAT*(poly(n))-hard. Given a random* 3AND *formula with n variables and $m = \Delta n$ clauses, output* typical *with probability at least $1/2$, but output* exceptional *on every formula with $(1/4 - \varepsilon)m$ simultaneously satisfiable clauses.*

We want to show that if we have some good approximation algorithm for BBIS in constant degree graphs, then we can use it to design a refutation algorithm for MAX-3AND, which contradicts Hypothesis 5.2.1. Before doing this, we introduce the following technical lemma, which states an upper bound on the probability of a random variable with bounded range falling far below its expectation, similar to the Markov inequality (see Appendix A.2).

Lemma 5.3.2. *Let $X \in [0, s]$ be a random variable with $\mathrm{E}[X] \geq \eta s$ for some $0 < \eta < 1$. Then*

$$\Pr\left(X \leq \frac{\eta s}{t}\right) \leq \frac{1-\eta}{1-\frac{\eta}{t}}$$

for any $t > 1$.

Proof: Towards a contradiction, assume that the claim does not hold. We may then write that

$$\begin{aligned}
\mathrm{E}[X] &\leq \Pr\left(X \leq \frac{\eta s}{t}\right) \cdot \frac{\eta s}{t} + \Pr\left(X > \frac{\eta s}{t}\right) \cdot s \\
&< \frac{1-\eta}{1-\frac{\eta}{t}} \cdot \frac{\eta s}{t} + \left(1 - \frac{1-\eta}{1-\frac{\eta}{t}}\right) s \\
&= \frac{\eta}{1-\frac{\eta}{t}} \left(\frac{1}{t} - \frac{\eta}{t} + 1 - \frac{1}{t}\right) s \\
&= \eta s,
\end{aligned}$$

a contradiction. □

Let us now have a closer look at the random 3AND formulas we are given as an input. Clearly, in expectation each literal will appear $(3/2)\Delta$ times in the formula. Now let V_i be a random variable counting the number of occurrences of literal ℓ_i. Applying the Chernoff bound (see Appendix A.2) we have that

$$\Pr\left[(1-\delta)\frac{3}{2}\Delta \leq V_i \leq (1+\delta)\frac{3}{2}\Delta\right] \geq 1 - 2\mathrm{e}^{-(3/4)\delta^2\Delta}$$

for any $0 < \delta < 1$. For every literal we define an additional random variable $X_i \in \{0, 1\}$ that indicates whether the above condition is satisfied and let $X = X_1 + \cdots + X_{2n}$. By linearity of expectation it obviously holds that

$$\mathrm{E}[X] \geq \left(1 - 2\mathrm{e}^{-(3/4)\delta^2\Delta}\right) 2n.$$

This implies that

$$\Pr\left[X < (1 - \sqrt{2}\mathrm{e}^{-(3/8)\delta^2\Delta})2n\right] \leq \sqrt{2}\mathrm{e}^{-(3/8)\delta^2\Delta},$$

by Lemma 5.3.2 with $\eta = 1 - 2\mathrm{e}^{-(3/4)\delta^2\Delta}$, $t = (1 - 2\mathrm{e}^{-(3/4)\delta^2\Delta})/(1 - \sqrt{2}\mathrm{e}^{-(3/8)\delta^2\Delta})$ and $s = 2n$. Now fix any $\gamma > 0$ and observe that by choosing Δ sufficiently large we can ensure that $\sqrt{2}\mathrm{e}^{-(3/8)\delta^2\Delta} \leq \gamma$.

Fact 5.3.3. *With probability* $1-\gamma$ *a* $(1-\gamma)$*-fraction of the literals appear between* $(1-\delta)\frac{3}{2}\Delta$ *and* $(1+\delta)\frac{3}{2}\Delta$ *times in a random* MAX-3AND *formula.*

The first step of our refutation algorithm for MAX-3AND consists of checking the above condition. If too many literals deviate from their expected number of occurrences, the algorithm outputs *exceptional*. If this is not the case, we continue by removing the few problematic literals from the formula. More precisely, we remove every clause that contains a literal appearing more than $(1+\delta)(3/2)\Delta$ times.

Let $\mu = 3(\delta+\gamma)$. We know that $(1-\gamma)2n$ *good* literals appear at least $(1-\delta)(3/2)\Delta$ times within the formula. Thus, a total number of at least

$$(1-\gamma)2n(1-\delta)(3/2)\Delta \geq (1-\delta-\gamma)3\Delta n$$

literal occurrences belong to *good* literals. This leaves at most $(\delta+\gamma)3\Delta n = \mu m$ literal occurrences belonging to *bad* literals and, consequently, gives an upper bound on the number of clauses that are removed from the formula. For the rest of the reduction to BBIS we need two more facts. Fact 5.3.5 is explicitly proven in [Fei02]. Fact 5.3.4 is immediate from the above.

Fact 5.3.4. *If the original* MAX-3AND *formula had* $(1/4-\varepsilon)m$ *satisfiable clauses, then the number of satisfiable clauses in our modified formula is bounded below by* $(1/4-\varepsilon-\mu)m$.

Fact 5.3.5. *For every* $\varepsilon > 0$, *sufficiently large* $\Delta \in \mathbb{N}$ *and* n *large enough, the following holds. With high probability every set of* $(1/8+\varepsilon)m$ *clauses in a random* MAX-3AND *formula with* $m = \Delta n$ *clauses contains at least* $n+1$ *different literals.*

We transform the modified formula into an instance of BBIS as follows. On both sides of the bipartition we have a vertex for every clause of the formula. Vertices on opposite sides are connected by an edge, if the corresponding clauses contain conflicting literals, i.e., if some variable appears in positive form in one clause and in negative form in the other. Thus, two vertices are connected if and only if the corresponding clauses cannot be satisfied simultaneously.

It is straightforward to argue that $(1/4-\varepsilon-\mu)m$ satisfiable clauses result in a balanced bipartite independent set of at least the same size, since for any given truth assignment we can select the vertices corresponding to satisfied clauses on both sides of the bipartition as a balanced bipartite independent set. On the other hand, for random formulas the size of the maximum balanced bipartite independent set is bounded above by $(1/8+\varepsilon)m$ with high probability, since by Fact 5.3.5 every selection of $(1/8+\varepsilon)m$ clauses contains at least $n+1$ distinct literals with high probability and, thus, is not satisfiable because one literal must appear in both positive and negative form. Additionally we know that, since every clause contains 3 literals and every literal appears at most $(1+\delta)(3/2)\Delta$ times, the resulting bipartite graph has a maximum degree of at most $(1+\delta)(9/2)\Delta$.

Assume now we had some polynomial time algorithm that can distinguish the two cases with an error probability exponentially close to 0. By applying this algorithm to the above BBIS instance we immediately obtain a polynomial time refutation algorithm for MAX-3AND with exponentially small failure probability for detecting exceptional formulas. If the BBIS algorithm returns a balanced bipartite independent set larger than $(1/8+\varepsilon)m$, we output *exceptional*. Otherwise, we output *typical*. The failure probability for detecting typical formulas is dominated by the probability that the formula has too many literals deviating from their expected number of occurrences and, thus, can be made an arbitrarily small constant. Hence, we have shown the following lemma.

Lemma 5.3.6. *Let $\mathcal{G}(a,d)$, $\mathcal{G}(b,d)$ be the families of bipartite graphs on $2n$ vertices with maximum degree bounded by $d \in \mathbb{N}$ and a maximum balanced bipartite independent set of size at most an or at least bn, respectively. There exist $0 < a < b < 1$ and $d \in \mathbb{N}$, such that deciding whether a given graph $G \in \mathcal{G}(a,d) \cup \mathcal{G}(b,d)$ belongs to $\mathcal{G}(a,d)$ or $\mathcal{G}(b,d)$ is R3SAT*(poly(n))-hard.*

5.3.2 Gap-Amplification for Bounded Degree BBIS

For a bipartite graph $G = (V, W, E)$, $|V| = |W| = n$, let $\alpha(G)$ refer to the size of a maximum balanced bipartite independent set in G. Let $\mathcal{G}(a,d)$ and $\mathcal{G}(b,d)$ be two families of bipartite graphs with maximum degree bounded by d and $\alpha(G) \le an$ for $G \in \mathcal{G}(a,d)$, $\alpha(G) \ge bn$ for $G \in \mathcal{G}(b,d)$. From the previous section we know that we can choose constants a, b and d, such that deciding whether a given graph is from $\mathcal{G}(a,d)$ or $\mathcal{G}(b,d)$ is hard assuming Hypothesis 5.2.1 holds. The following definition is in analogy to Definition 2.3.3.

Definition 5.3.7. *Let $G = (V, W, E)$, $|V| = |W| = n$, be a bipartite graph and $k \in \mathbb{N}$. The k*-fold graph product *$G^k = (V^k, W^k, E_k)$ is defined by Cartesian products V^k, W^k and $\{(v_1, \dots, v_k), (w_1, \dots, w_k)\} \in E_k$ if and only if $\{v_1, \dots, v_k, w_1, \dots, w_k\}$ is not a bipartite independent set in G.*

We briefly describe the application of derandomized graph products [AFWZ95] to bipartite graphs. Given $G = (V, W, E)$, $|V| = |W| = n$, we construct a non-bipartite δ-regular Ramanujan graph H on n vertices with constant degree δ (depending only on constants a and b). Vertices V^k and W^k of the derandomized graph product DG^k are obtained by choosing a vertex of H uniformly at random and taking a random walk of length $k-1$ starting at this vertex. For $k = O(\log n)$ the number $n\delta^{k-1}$ of such random walks is polynomial and, thus, DG^k can be constructed deterministically in polynomial time. The edges of DG^k are defined as before.

An analysis similar to the one in the proof of Theorem 2.3.5 yields Theorem 5.2.5. In contrast to the IS case, Theorem 5.2.5 is formulated in terms of a gap version of BBIS. Note, that it is possible to determine the necessary values of $a(n)$ and $b(n)$ from the proof of Theorem 2.3.5. We want to remark that by construction the constant degree graphs obtained by the reduction in Section 5.3.1 are symmetric in the sense that we can rename vertices $V = \{v_1, \dots, v_n\}$ and $W = \{w_1, \dots, w_n\}$, such that $\{v_i, w_j\} \in E$ if and only if $\{v_j, w_i\} \in E$. This property is not lost during gap amplification, since we can use the same expander graph to obtain the vertices on both sides of the graph product.

5.3.3 Maximum Expanding Sequences

Let $G \in \mathcal{G}(a(n), f(n)) \cup \mathcal{G}(b(n), f(n))$, $G = (V, W, E)$,$|V| = |W| = n$, with $a(n)$, $b(n)$ and $f(n)$ as in Theorem 5.2.5 be given. We will reduce the problem of deciding whether $G \in \mathcal{G}(a(n), f(n))$ or $G \in \mathcal{G}(b(n), f(n))$ to solving a restricted instance of MES. We start by adding a couple of random edges to the graph. More precisely, every possible edge is added to G with probability $(b(n)n)^{-1}$. We do not allow multiple edges and, thus, edges that have already been present in G will not be duplicated.

Afterwards we remove vertices whose degree has become too high from the graph. In expectation the random experiment tries to add $b(n)^{-1}$ new edges to every vertex $v \in V \cup W$. We remove a vertex v

and all its incident edges if more than $c \cdot b(n)^{-1}$ edges are added to it, where c is some sufficiently large constant to be determined later. Let A_v be the random variable counting the number of edges added to v. Applying the Chernoff bound (see Appendix A.2) we obtain

$$\Pr(v \text{ is removed}) = \Pr(A_v \geq c \cdot b(n)^{-1}) \leq \mathrm{e}^{-c/b(n)}$$

for any constant $c \geq 3\mathrm{e} - 1$. We denote the modified graph by $G' = (V', W', E')$. For every vertex $v_i \in V'$ we define a corresponding set S_i by

$$S_i = \left\{ w_j \in W' \,|\, \{v_i, w_j\} \in E' \right\},$$

i.e., vertices V' will correspond to sets over universe W' in our MES instance. In order to obtain a feasible MES instance we need to define an order on sets S_i, which we do next. Observe that vertices in G' have degree at most

$$f'(n) \leq f(n) + c \cdot b(n)^{-1} = \mathcal{O}(f(n)),$$

where we use the fact that bipartite graphs with bounded degree $f(n)$ have a balanced bipartite set of size at least $n/(f(n)+1)$ and, thus, it must be the case that $b(n)^{-1} = \mathcal{O}(f(n))$. Furthermore, if the maximum degree of G' is $f'(n)$, then the sets S_i can be partitioned into $f'(n)^2$ many classes, such that sets in each class do not intersect. To see this, note, that every set contains at most $f'(n)$ elements, each of which is contained in at most $f'(n) - 1$ further sets. Thus, starting with $f'(n)^2$ empty classes and adding sets one by one, the number of classes to which a specific set cannot be added is always bounded above by $f'(n)(f'(n) - 1) < f'(n)^2$.

Let $\mathcal{C}_1, \ldots, \mathcal{C}_\kappa$ denote the classes of sets obtained in this way and observe that $\kappa = \mathcal{O}(f(n)^2)$. We reorder sets according to the classes and finally obtain an MES instance $S_1, \ldots, S_m$ for which it holds that $\mathcal{C}_j = \{S_{k(j)}, S_{k(j)+1} \ldots S_{\ell(j)}\}$, i.e., sets belonging to a single class form a non-interrupted block in the ordering. Thus, the MES-instance is $f(n)^2$-separable (see Definition 5.2.8). This property is not required for the remainder of this section, but will be of immense importance for the reduction to uniform-budget UDP($\mathcal{C}$)-MIN in Section 5.

Soundness: Let $G \in \mathcal{G}(b(n), f(n))$. Assume for the moment that no vertices are removed from G and let $\mathcal{S}^* = \{S_{\phi(1)}, \ldots, S_{\phi(\ell)}\}$, $\ell = \lceil b(n)n \rceil$, be the sets in the MES instance corresponding to vertices from V that belong to a maximum balanced bipartite independent set. Analogously, let $W^* \subset W$ denote the vertices from W belonging to the balanced bipartite independent set. For $1 \leq j \leq \ell/2$ consider set $S_{\phi(j)}$. We say that $S_{\phi(j)}$ is *successful* if we can use it to construct a large expanding sequence or, more formally, if conditions A_j through D_j below are satisfied. When we analyze the success probability of set $S_{\phi(j)}$, a subtle issue that we need to address is the fact that we have reordered the sets in order to obtain a separable instance. We resolve this problem by proving a lower bound on the success probability that holds with respect to any reordering.

- A: Condition A_j is satisfied if $|S_{\phi(j)} \cap W^*| = 1$.
- B: Condition B_j is satisfied if $S_{\phi(j)} \cap S_{\phi(i)} \cap W^* = \emptyset$ for all $1 \leq i \leq \ell/2$, $i \neq j$.
- C: Condition C_j is satisfied if the vertex corresponding to set $S_{\phi(j)}$ is not removed due to the degree constraint.

D: Condition D_j is satisfied if none of the vertices in $S_{\phi(j)} \cap W^*$ are removed due to the degree constraint.

It is not difficult to check that successful sets belong to the MES-instance and form an expanding sequence, since their corresponding vertices are not removed from the graph and each set covers a unique element in W^*, which yields the necessary expansion property. Let us now determine the probability that set $S_{\phi(j)}$ is successful. We can write that

$$\begin{aligned}\Pr(S_{\phi(j)} \text{ is successful}) &= 1 - \Pr(\overline{A}_j \vee \overline{B}_j \vee \overline{C}_j \vee \overline{D}_j)\\ &= 1 - \Pr(\overline{A}_j)\\ &\quad - \Pr(\overline{B}_j|A_j)\cdot\Pr(A_j)\\ &\quad - \Pr(\overline{C}_j|A_j \wedge B_j)\cdot\Pr(A_j \wedge B_j)\\ &\quad - \Pr(\overline{D}_j|A_j \wedge B_j \wedge C_j)\cdot\Pr(A_j \wedge B_j \wedge C_j).\end{aligned}$$

We first consider event A_j and obtain

$$\begin{aligned}\Pr(A_j) &= \sum_{w\in W^*}\Pr\big(S_{\phi(j)}\cap W^* = \{w\}\big) = \sum_{w\in W^*}\frac{1}{\lceil b(n)n\rceil}\left(1-\frac{1}{b(n)n}\right)^{\lceil b(n)n\rceil-1}\\ &\approx b(n)n\frac{1}{\mathrm{e}b(n)n} = \frac{1}{\mathrm{e}},\end{aligned}$$

where the above holds with arbitrary precision for large values of n. Let us then consider $\Pr(\overline{B}_j|A_j)$. Sets $S_{\phi(j)}$ and $S_{\phi(i)}$ contain every element from W^* with equal probability $1/b(n)n$. Furthermore, $S_{\phi(j)} \cap W^*$ and $S_{\phi(j)} \cap W^*$ are independent by construction. Thus,

$$\Pr(\overline{B}_j|A_j) \le \sum_{i=1}^{\ell/2}\sum_{w\in W^*}\Pr(w\in S_{\phi(j)})\cdot\Pr(w\in S_{\phi(i)}) \le \frac{b(n)n}{2}b(n)n\frac{1}{(b(n)n)^2} = \frac{1}{2}.$$

We have already seen that the probability of any specific vertex being removed due to the degree constraint is bounded above by $\mathrm{e}^{-c/b(n)}$. We conclude that

$$\Pr(\overline{C}_j|A_j\wedge B_j)\cdot\Pr(A_j\wedge B_j) \le \Pr(\overline{C}_j) \le \mathrm{e}^{-c/b(n)},$$

and the same estimate obviously holds for $\Pr(\overline{D}_j|A_j \wedge B_j \wedge C_j)\cdot\Pr(A_j \wedge B_j \wedge C_j)$. Finally, this yields

$$\begin{aligned}\Pr(S_{\phi(j)} \text{ is successful}) &\ge 1-\left(1-\frac{1}{\mathrm{e}}\right)-\frac{1}{2}\frac{1}{\mathrm{e}}\\ &\quad -\mathrm{e}^{-c/b(n)}-\mathrm{e}^{-c/b(n)}\\ &\ge \frac{1}{2\mathrm{e}}-2\mathrm{e}^{-c/b(n)} \approx \frac{1}{2\mathrm{e}}\end{aligned}$$

for sufficiently large constant c. Let Y denote the number of successful sets. It obviously holds that $\mathrm{E}[Y] \ge (1/4\mathrm{e})b(n)n$ and applying Lemma 5.3.2 with $\eta = 1/4\mathrm{e}$, $t = 2$ and $s = b(n)n$ we get that

$$\Pr\big(Y \le \frac{1}{8\mathrm{e}}b(n)n\big) \le 1-\frac{1}{8\mathrm{e}}.$$

This implies that with probability $\Omega(1)$ there exists an expanding sequence of length $\Omega(b(n)n)$.

Completeness: Let $G \in \mathcal{G}(a(n), f(n))$ and consider any expanding sequence ϕ in $S_1, \ldots, S_m$. Since the maximum balanced bipartite independent set in G is of size $a(n)n$, every selection of $a(n)n + 1$ vertices from V must be adjacent to all but $a(n)n$ vertices from W. Thus, the first $a(n)n + 1$ sets from ϕ leave at most $a(n)n$ elements uncovered. Since the expansion property requires that every further set in the sequence must contain a previously uncovered element, it follows that $|\phi| \le 2a(n)n + 1$.

We have shown a randomized reduction with constant one-sided error probability. By repeating the algorithm a polynomial number of times, we obtain error probabilities that are exponentially close to 0. This proves Lemma 5.2.9.

5.3.4 Reduction to UDP($\mathcal{C}$)-MIN

The final step in the proof of Theorem 5.2.10 consists of reducing $f(m)$-separable MES to UDP($\mathcal{C}$)-MIN with uniform budgets. Let MES instance $S_1, \ldots, S_m$ be separable into $\mathcal{C}_1, \ldots, \mathcal{C}_\kappa$ with $\kappa = \mathcal{O}(f(m))$.

For each element e in the universe of the MES instance we have a corresponding product u_e. For every set S_i in class $\mathcal{C}_k$ we define a collection of 2^{k-1} identical consumers $C_i = \{c_i^1, c_i^2, \ldots, c_i^{2^{k-1}}\}$. Each of these consumers has budget $b_i = 2^{1-k}$ and is interested in products u_e corresponding to elements $e \in S_i$. Note, that the total number of consumer samples in this construction is bounded above by $n2^{\mathcal{O}(f(n))}$.

Soundness: Let $\phi = (\phi(1) < \cdots < \phi(\ell))$ be an expanding sequence of length ℓ. For every $1 \le i \le \ell$ let $N_{\phi(i)}$ denote the elements that are newly covered by $S_{\phi(i)}$. Now we repeat the following for $i = 1, \ldots, \ell$. Determine $N_{\phi(i)}$, then set the prices of all products u_e corresponding to some $e \in N_{\phi(i)}$ to b_i. For consumers $C_{\phi(i)}$ it then holds that $p(u_e) = b_i$ for all $e \in N_{\phi(i)}$, $p(u_e) > b_i$ for all $e \in S_{\phi(i)} \backslash N_{\phi(i)}$. As a result, all 2^{k-1} consumers belonging to a set $S_{\phi(i)}$ in the expanding sequence will buy at their budget value $b_i = 2^{1-k}$ and contribute profit 1. Thus, overall profit from consumers corresponding to the expanding sequence is at least ℓ. An illustration of this construction is found in Fig. 5.2.

Completeness: Assume that we are given a price assignment resulting in overall revenue r. First observe that w.l.o.g. all prices are from the set of distinct budget values, i.e., all prices are powers of 2. Then note that w.l.o.g. revenue at least $r/2$ is due to consumers buying at their budget values, since otherwise we could increase overall revenue by multiplying all prices by 2. Finally, it's not difficult to see that consumers buying at their budget values form an expanding sequence. It follows that we obtain an expanding sequence ϕ of length at least $r/2$. This finishes the proof of Theorem 5.2.10.

Finally, we briefly discuss Theorem 5.2.11. R3SAT*($2^{\mathcal{O}(n^\delta)}$)-hardness of approximating uniform-budget UDP($\mathcal{C}$)-MIN within $\mathcal{O}(n^\varepsilon)$ follows immediately by choosing $f(m) = m^\varepsilon$ for arbitrarily small $\varepsilon > 0$.

To obtain inapproximability for instances with a bounded number of non-zero budgets per consumer we have to start from BBIS in constant degree graphs again. As shown in [AFWZ95] for the independent set problem, BBIS in graphs of degree at most Δ is R3SAT*(poly(n))-hard to approximate within a factor of Δ^ε for some $\varepsilon > 0$ and all $\Delta \ge \Delta_0$, where Δ_0 is constant d from Lemma 5.3.6. We then apply our reduction as described above and obtain uniform-budget UDP($\mathcal{C}$)-MIN instances with $\ell = \Delta$ non-zero budgets per consumer and inapproximability within $(1/(16\mathrm{e}))\Delta^\varepsilon$, where the factor $1/16\mathrm{e}$ stems from the fact that the randomized reduction from BBIS to MES might blow up small independent sets by a factor of 2 or shrink large independent sets by a factor of $1/8\mathrm{e}$. Choosing $\ell_0 \in \mathbb{N}$ sufficiently large ensures that $\ell^{\varepsilon-\delta} \leq (1/(16\mathrm{e}))\ell^{\varepsilon}$ for all $\ell \geq \ell_0$.

5.4 Literature

The max-gain selection rule for unit-demand pricing with unlimited product supply was originally proposed in [AFMZ04]. Guruswami et al. [GHK$^+$05] first considered the limited-supply version, which they termed unit-demand envy-free pricing. They show that the problem allows approximation guarantees that are logarithmic in the number of consumers and prove APX-hardness. Hartline and Koltun [HK05] consider the special case of max-gain and envy-free pricing with only a constant number of different products. They derive fully polynomial time approximation schemes that run in near-linear time in the max-gain and near-cubic time in the more involved limited-supply envy-free case. Chawla et al. [CHK07] investigate the special case of distribution-based max-gain buying, in which consumers are drawn from a product distribution, i.e., their budgets for different products are chosen independently. They prove that in this situation constant approximation guarantees are possible using concepts from the theory of optimal auction design [Mye81]. Krauthgamer et al. [KMR07] consider so-called bimodal markets, in which each consumer's budget is chosen from a set of only two possible alternatives $\{1, C\}$, and show that LP-based randomized rounding techniques yield approximation guarantee $2 - 1/\ell - (\ell - 1)/(\ell C)$ when consumers have at most ℓ non-zero budgets.

The first result for general BBIS using a quite moderate complexity theoretic assumption was obtained by Khot [Kho04]. Previous results by Feige [Fei02] and Feige and Kogan [FK04] are deriving hardness of BBIS under more specific assumptions. The connection between average-case hardness of random 3SAT and BBIS is made in [Fei02].

The technique of scaling the hardness of some given base problem to allow sampling-based encoding in terms of revenue maximizing pricing stems from [BK07]. The first connection between BBIS and a related pricing problem (see Chapter 6) was made by Demaine et al. [DFHS06]. The results from this Chapter have been published in [Bri06].

6 Network Pricing I: The Single-Minded Pricing Problem

After our investigation of various aspects of unit-demand pricing, we will now turn to another natural and well-studied pricing problem. In the *Single-Minded Pricing Problem*, which was first considered in [GHK$^+$05], we assume that products are *pure complements* rather than pure substitutes, i.e., each consumer is interested in purchasing a subset of the available products and will purchase all of them or none at all, depending on whether her budget constraint is violated by the sum of product prices.

This model of consumer behavior is particularly popular in the context of algorithmic mechanism design, where a lot of work has been done regarding the design of truthful auctions among single-minded agents. Consequently, similar interest has been paid to the corresponding pricing problem because of its intrinsic connection to the problem of revenue maximization in strategic settings.

As far as its general approximability is concerned, single-minded pricing behaves quite similar to unit-demand pricing. Maybe not very surprisingly, the single-price algorithm turns out to be applicable once more and yields approximation guarantees that are logarithmic in the number of consumers and products. On the other hand, single-minded pricing was the first problem from the realm of pricing for which super-constant and, most importantly, essentially tight inapproximability results could be proven [DFHS06]. We will briefly review these results and point out that the techniques from Chapter 5 yield an interesting supplement to the known approximation thresholds.

However, for the major part of this chapter we will take a different point of view on the problem. Assume that instead of pricing abstract products, we are given a network in which we may assign prices to the edges and consumers aim to purchase fixed paths connecting their terminal pairs. As one possible application, we can think of the underlying graph as a public transportation network in which we want to price railroad or flight connections. Similarly, we might be faced with a computer network in which a number of service providers need to purchase backbone connections between different sites. In both cases, one might hope that the problem exhibits some kind of structure that allows for better approximation guarantees than are possible in general. In transportation networks, we would expect the length of the requested paths to be bounded, since consumers are not interested in connections that result in too many stopovers. If we price high capacity network links for service providers, we might hope that the number of requests per link is not too large. Apart from this, we could restrict ourselves to cases where the underlying network itself is rather sparse.

The results in this chapter are twofold. First, we will see that single-minded pricing remains quite hard, even if we restrict ourselves to the network setting, and even if we require problem instances to be sparse in various aspects. On the algorithmic side, we show that improved approximation guarantees are still possible in some cases. First, we present an FPTAS for the case that the sets desired by consumers have a specific structure. We then present an algorithm for the general problem whose approximation guarantee asymptotically matches the ratio of the single-price algorithm in the worst case, yet, which is capable of exploiting the sparse problem structure of instances in which both the number of products a consumer is interested in and the number of consumers interested in a specific product are bounded in order to derive improved ratios.

The rest of this chapter is organized as follows. A formal definition of the single-minded pricing problem is found in Section 6.1. Section 6.2 presents some results on the approximability of general SMP. Section 6.3 deals with the highway problem, the variation of G-SMP in which the underlying graph is simply a line. We first give a proof of NP-completeness and then show how to derive an FPTAS for the restricted case used for the preceding hardness result. Section 6.4 proves APX-hardness of single-minded pricing in networks which hold even under various strong restrictions. Section 6.5 presents the approximation algorithm for the problem. An overview of related literature is found in Section 6.6.

6.1 Preliminaries

The unlimited-supply single-minded pricing problem (SMP) has first been formulated in [GHK$^+$05]. Intuitively, a consumer specifies the maximum price she is willing to pay for a certain set of products. After a price has been assigned to each product, consumers decide to buy their sets depending on whether the sum of prices of goods contained in their sets exceeds their specified budget. As in the case of unit-demand pricing with uniform budgets we will associate a single-minded consumer with the set she is interested in and mostly talk of a *set* and its *value* rather than a consumer and her budget.

Definition 6.1.1. *Given a universe $\mathcal{U}$, $|\mathcal{U}| = n$, of products and a collection $\mathcal{S}$, $|\mathcal{S}| = m$, of subsets of $\mathcal{U}$ with associated values $v(S)$ for all $S \in \mathcal{S}$, the unlimited-supply* single-minded pricing problem *(*SMP*) asks for prices $p : \mathcal{U} \to \mathbb{R}_0^+$ maximizing*

$$r(p) = \sum_{S \in \mathcal{A}(p)} \sum_{u \in S} p(u),$$

where $\mathcal{A}(p) = \{S \in \mathcal{S} \mid \sum_{u \in S} p(u) \leq v(S)\}$.

By $\delta(S) = v(S)/|S|$ we will refer to the *price per item* of set S. In case of single-minded network pricing, we assign prices to the edges of a graph and consumers want to connect terminal sets through fixed paths, which they purchase if the sum of prices of edges on their path does not exceed their budgets. Whenever the problem is defined on an underlying graph, we slightly adjust our notation to reflect this difference.

Definition 6.1.2. *In the* SMP problem on graphs *(*G-SMP*) we are given a graph $G = (V, E)$ rather than a universe of products and a collection $\mathcal{P}$ of paths in G with associated values $v(P)$ for all $P \in \mathcal{P}$. We want to find revenue maximizing prices $p : E \to \mathbb{R}_0^+$.*

As mentioned before, G-SMP nicely models the pricing of direct connections in public transportation networks and has therefore been termed *tollbooth problem* in [GHK$^+$05]. If the underlying graph is simply a line, we can think of this so-called *highway problem* as pricing segments of some privately owned highway. In analogy to general SMP we denote $|E| = n$, $|\mathcal{P}| = m$.

6.2 General Approximability

As mentioned, a lot of work dealing with single-minded pricing and its various applications has been done. We will briefly summarize the most important results concerning the problem's approximability. Guruswami et al. [GHK$^+$05] show that the single-price algorithm can be applied to SMP, choosing as candidate prices $\delta(S)$ for all $S \in \mathcal{S}$.

Theorem 6.2.1 ([GHK+05]). *The single-price algorithm computes an $(H_m + H_n)$-approximation with respect to optimal revenue for* SMP. *This bound is tight.*

Balcan and Blum [BB06] show that the random-partitioning algorithm (Algorithm 2) achieves approximation guarantee $\mathcal{O}(\ell)$ on instances in which the maximum cardinality of any set is bounded by ℓ.

Theorem 6.2.2 ([BB06]). SMP *with sets of maximum cardinality ℓ can be approximated in polynomial time within $\mathcal{O}(\ell)$.*

Demaine et al. [DFHS06] present the first super-constant lower bound on the approximability of any pricing problem and prove a near-tight lower bound for approximation guarantees expressed in terms of the number m of consumers.

Theorem 6.2.3 ([DFHS06]). SMP *is hard to approximate within $\mathcal{O}(\log^{\varepsilon} m)$ for some constant $\varepsilon > 0$, unless* NP $\subseteq \bigcap_{\delta>0}$ BPTIME($2^{\mathcal{O}(n^{\delta})}$).

For unit-demand pricing we have seen that we could actually achieve asymptotically better lower bounds for approximation ratios parametrized in terms of the number of products n. The reduction that is used to prove Theorem 6.2.3 above yields instances in which $v(S) = 1$ for every set S. Consequently, it is not possible to prove better than logarithmic bounds based on this construction. To see this, note, that we can easily achieve approximation guarantee $\mathcal{O}(\log n)$ on this type of instance. We simply partition sets into a $\mathcal{O}(\log n)$ subclasses $\mathcal{S}_k$ according to their cardinality. For sets $S \in \mathcal{S}_k$ it holds that $2^k \leq |S| < 2^{k+1}$ and by setting $p(u) = 1/2^{k+1}$ we can realize revenue at least $(1/2)|\mathcal{S}_k|$ from this subclass of sets.

Interestingly, our results from Chapter 5 yield an easy way to close this gap. More formally, with some slight modifications the reduction from MES to uniform-budget UDP($\mathcal{C}$)-MIN in Section 5.3.4 works for SMP, as well, and we obtain the following result in analogy to Theorem 5.2.11. Note, that the lower bounds in Theorem 6.2.4 are essentially tight by Theorem 6.2.2.

Theorem 6.2.4. *There exist constants $\ell_0 \in \mathbb{N}$ and $\varepsilon > 0$, such that for every $\ell \geq \ell_0$ it is R3SAT*(poly(n))-hard to approximate* SMP *with sets of maximum cardinality ℓ within ℓ^{ε}. Furthermore, for every $\delta > 0$ there exists $\varepsilon > 0$, such that it is R3SAT*($2^{\mathcal{O}(n^{\delta})}$)-hard to approximate* SMP *within $\mathcal{O}(n^{\varepsilon})$. Hardness of approximation holds even under the weaker assumption of Hypothesis 5.2.4.*

Proof: We prove the second part of the theorem. Let a given MES instance $S_1, \ldots, S_m$ be separable into $\mathcal{C}_1, \ldots, \mathcal{C}_\kappa$ with $\kappa = \mathcal{O}(f(m))$. For each element e in the universe of the MES instance we have a corresponding product e. We assume that the MES instance is defined on a universe of size n and $n = m$. This is w.l.o.g. due to the reduction in Section 5.3.3. For every set S_i in class $\mathcal{C}_j$ we define a collection of $(2n)^{\kappa-j}$ identical sets $\mathcal{S}_i = \{S_i^1, S_i^2, \ldots, S_i^{2^{\kappa-j}}\}$. Each of these sets is an exact copy of S_i, i.e., $S_i^k = S_i$ for all k. Furthermore, we define $v_i = (2n)^{j-\kappa}$ and let $v(S_i^k) = v_i$ for all sets S_i^k. Note, that the total number of sets in this construction is bounded above by $n(2n)^{\mathcal{O}(f(n))}$.

Soundness: Let $\phi = (\phi(1) < \cdots < \phi(\ell))$ be an expanding sequence of length ℓ. For every $1 \leq i \leq \ell$ let $N_{\phi(i)}$ denote the elements that are newly covered by $S_{\phi(i)}$. Now we repeat the following for $i = 1, \ldots, \ell$. Determine $N_{\phi(i)}$ and let

$$\xi_{\phi(i)} = \sum_{e \in S_{\phi(i)} \setminus N_{\phi(i)}} p(e)$$

denote the sum of prices of previously covered elements contained in $S_{\phi(i)}$. Let $S_{\phi(i)} \in \mathcal{C}_j$ By the fact that sets from class $\mathcal{C}_j$ do not intersect and budgets increase exponentially between different classes, it follows that every $e \in S_{\phi(i)} \backslash N_{\phi(i)}$ has price $p(e) \leq v_{\phi(i)}/(2n)$ and we get that $\xi_{\phi(i)} < v_{\phi(i)}/2$. Thus, $v_{\phi(i)} - \xi_{\phi(i)} > 0$ and by setting the prices of all $e \in N_{\phi(i)}$ to

$$p(e) = \frac{v_{\phi(i)} - \xi_{\phi(i)}}{|N_{\phi(i)}|}$$

we extract revenue $|\mathcal{S}_{\phi(i)}|v_{\phi(i)} = (2n)^{\kappa-j}(2n)^{j-\kappa} = 1$ from sets $\mathcal{S}_{\phi(i)}$. Consequently, overall revenue is at least ℓ.

Completeness: Assume that we are given a price assignment p resulting in overall revenue r. W.l.o.g. revenue at least $r/2$ is due to consumers buying at a price that is at least half their budget value, since otherwise we could increase overall revenue by multiplying all prices by 2.

Consider set S_i. For sets S_k with $S_i \cap S_k \neq \emptyset$ and $k < i$ it must be true that $v_k \leq v_i/(2n)$. Thus, if S_i is contained in the union of sets with smaller indices whose corresponding sets in the SMP instance yield positive revenue, it follows that $p(e) \leq v_i/(2n)$ for all $e \in S_i$. Consequently, revenue from any set S_i^k in the SMP instance is at most $f(n)v_i/(2n) < v_i/2$. Conversely, sets that yield revenue equal to at least half their budget values must adhere to the expansion property and we obtain an expanding sequence ϕ of length at least $r/2$.

Finally, for any given value of $\delta > 0$ fix $\delta' < \delta$ and let $f(n) = n^{\delta'}$. We obtain MES instances of size $2^{\mathcal{O}(n^\delta)}$, which are hard to approximate within $\mathcal{O}(n^\varepsilon)$ for some $\varepsilon > 0$. The first part of the theorem follows similar to our argumentation in Section 5.3.4. □

Analogous lower bounds can also be obtained for the distribution-based version of SMP. However, as we shall be mainly interested in G-SMP and other types of sparse problem instances, we do not formally state these results at this point.

6.3 The Highway Problem

We start by considering the special case of G-SMP in which the underlying graph structure is simply a line, usually referred to as the highway problem. Guruswami et al. [GHK+05] give a pseudopolynomial time algorithm, which can be turned into an FPTAS by standard scaling and rounding techniques, for the case in which the length of all paths is bounded by a constant. We introduce another special case of the problem in which we allow paths to have arbitrary lengths but require that they are *nested*, i.e., given any two paths $P_1, P_2 \in \mathcal{P}$ we have that $P_1 \subseteq P_2$, $P_2 \subseteq P_1$ or $P_1 \cap P_2 = \emptyset$. Note, that if sets are nested it does not make a difference if the problem is defined on a graph or not. In fact, each such SMP instance can be viewed as being defined on a line by simply ordering the goods appropriately. We will prove NP-hardness of this problem and show how to derive an FPTAS by a dynamic programming approach.

6.3.1 NP-Hardness

Theorem 6.3.1 proves NP-hardness of the highway problem with nested paths. Hardness holds even if we do not allow multi-paths, i.e., multiple identical paths in the instance. Interestingly, our reduction yields instances that contain only a single *long* path, with the length of all other paths being at most 2. So one could say that it is only a single path that separates us from a matching hardness result for the FPTAS from [GHK+05] for the case of constant length paths.

Theorem 6.3.1. *G-SMP with nested paths (even without multi-paths) is NP-hard.*

Proof: We are going to prove NP-hardness by a reduction from the PARTITION problem. Given weights $w_1, \ldots, w_n \in \mathbb{R}^+$ we want to find a set $S \subset \{1, \ldots, n\}$, such that $\sum_{i \in S} w_i = \sum_{i \notin S} w_i$, i.e., find a partitioning into two sets of identical total weight. PARTITION is known to be NP-hard [GJ79].

For each weight w_i we construct a *weight gadget* W_i as depicted in Figure 6.1(a). On a graph consisting of 3 vertices v_1^i, v_2^i, v_3^i and 2 edges e_1^i, e_2^i we define the 3 possible different paths $P_1^i = \{e_1^i\}$, $P_2^i = \{e_2^i\}$, $P_3^i = \{e_1^i, e_2^i\}$ and let $v(P_1^i) = v(P_2^i) = v(P_3^i) = w_i$. We start with a simple observation about weight gadgets which turn out to ensure half-integrality of any locally optimal price assignment.

Fact 6.3.2. *The maximum revenue obtainable from weight gadget W_i is $2w_i$. If revenue $2w_i$ is obtained under price assignment p then $p(e_1^i) = p(e_2^i) = w_i$ or $p(e_1^i) + p(e_2^i) = w_i$.*

Fact 6.3.2 can be seen as follows. If path P_3^i contributes to the revenue, it must be the case that $p(e_1^i) + p(e_2^i) \leq w_i$. P_1^i and P_2^i can never give more revenue than $p(e_1^i)$ and $p(e_2^i)$, respectively. It immediately follows that total revenue is at most $2w_i$. On the other hand, the revenue obtained from each P_1^i and P_2^i is also bounded by w_i and so revenue $2w_i$ can also not be exceeded if P_3^i does not contribute. This gives the first part of the claim. From the above argumentation it follows that revenue $2w_i$ cannot be reached if $p(e_1^i) + p(e_2^i) < w_i$, while, e.g., $p(e_1^i) = p(e_2^i) = w_i/2$ results in full revenue. Now assume that p results in revenue $2w_i$ but $p(e_1^i) + p(e_2^i) > w_i$. Then P_3^i obviously contributes 0. It follows that P_1^i and P_2^i must give revenue w_i each and, thus, $p(e_1^i) = p(e_2^i) = w_i$.

We note that we can make use of the symmetry of a weight gadget by slightly changing the above claim and requiring w.l.o.g. that $p(e_1^i) = p(e_2^i) = w_i/2$ instead of $p(e_1^i) + p(e_2^i) = w_i$. In fact, we will from now on only consider the price $p(W_i) = p(e_1^i) + p(e_2^i)$ that is assigned to weight gadget W_i and implicitly assume that this price is split evenly among edges e_1^i and e_2^i. Revenue $r(W_i)$ from weight gadget W_i as a function of $p(W_i)$ is depicted in Fig. 6.1(b).

We now define the final G-SMP instance. The weight gadgets $W_1, \ldots, W_n$ are assembled into a single line by identifying vertices v_3^i and v_1^{i+1} for $i = 1, \ldots, n-1$. We define one more path P running all the way from v_1^1 to v_3^n and set $v(P) = (3/2) \sum_{i=1}^n w_i$, as shown in Figure 6.1(a). It is then straightforward to argue that total revenue $(7/2) \sum_{i=1}^n w_i$ can be reached on this instance if and only if a partitioning S with $\sum_{i \in S} w_i = \sum_{i \notin S} w_i$ exists. For the one direction we define prices p by $p(W_i) = 2w_i$ if $i \in S$ and $p(W_i) = w_i$ else. For the other direction one argues that the optimal pricing is of just this form and can choose $S = \{i \,|\, p(W_i) = 2w_i\}$ as the desired partitioning. □

We next take a look at the nested paths case from the algorithmic side.

6.3.2 An FPTAS

We present a pseudopolynomial time algorithm for SMP with nested paths which can be turned into an FPTAS by scaling and rounding the input appropriately. For the description of the dynamic programming

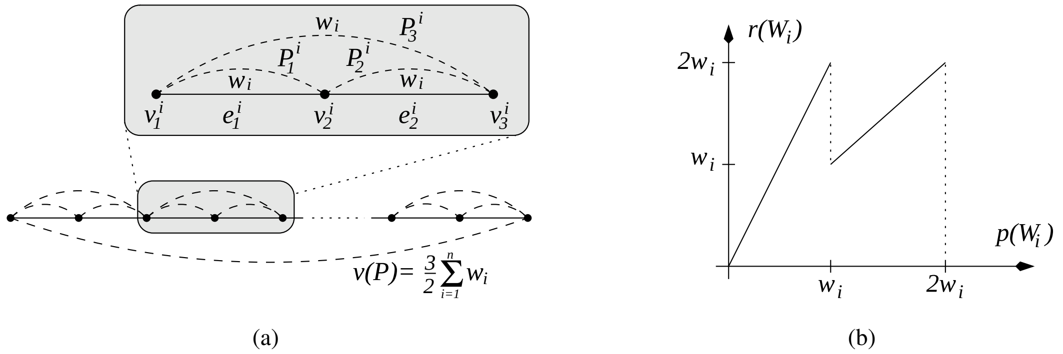


Figure 6.1: (a) A single weight gadget W_i and path P ensuring that maximum revenue can be reached only if the weights can be 2-partitioned. (b) Revenue from weight gadget W_i as a function of its total price $p(W_i)$.

approach we assume that all declared prices are integral. By an observation in [GHK$^+$05] this guarantees the existence of an integral optimal solution. To avoid technicalities, we assume here that every path is requested only once, i.e., there are no multi-paths. It is, however, not difficult to extend the algorithm to incorporate this more general case. Given a G-SMP instance with edges E and nested paths $\mathcal{P}$, we define the set of *intervals* $\mathcal{I}$ as follows. First, each path P defines an interval $I_P = P$. Then we add the interval $I_E = E$ containing all edges. Now consider any interval I and let J be the maximum size interval contained in I. We then also add $K = I \backslash J$ to $\mathcal{I}$ if it is not already contained. Note, that in general K need not be an interval in the classical sense, since it might contain edges that are situated on the left or right of J, respectively. It is, however, w.l.o.g. to assume that this is not the case, because we can always reorder edges to ensure that I and J have the same left border. Conceptually, at this point we view paths as simply collections of edges and observe that edges can be arranged in a single line, such that paths are mapped onto intervals. If interval I_P is defined by path P we let $v(I_P) = v(P)$ be the interval's value. If I is defined in a later step (i.e., not defined by a path), we set $v(I) = 0$. Note, that intervals can naturally be arranged as a binary tree with root corresponding to the complete line. For any interval I we let A_b^I refer to the maximum revenue that can be obtained from paths fully contained in I under the condition that the prices assigned to edges in I sum up to exactly b, i.e., $\sum_{e \in I} p(e) = b$. Consider interval I containing maximum length subintervals J and K. We have

$$A_b^I = \begin{cases} \max_{b'} A_{b'}^J + A_{b-b'}^K + b, & \text{if } b \leq v(I) \\ \max_{b'} A_{b'}^J + A_{b-b'}^K, & \text{else} \end{cases}$$

by the observation that any path which is contained in I is also fully contained in either J or K. For all minimal intervals I not containing any subintervals it trivially holds that $A_b^I = b$ if $b \leq v(I)$ and $A_b^I = 0$ else. We now only need to compute $\max_b A_b^{I_E}$ to find the optimal price assignment by simple backtracking.

Lemma 6.3.3. *The above algorithm finds an optimal solution for any instance of G-SMP with nested paths and integral valuations in time $O(n^3 B^2)$, where $B = \max_{P \in \mathcal{P}} v(P)$.*

The pseudopolynomial time algorithm can easily be turned into an FPTAS by standard rounding techniques. To this end, let $\alpha = nm/(\varepsilon B)$ and define scaled maximum prices $v'(P) = \lfloor \alpha v(P) \rfloor$ for all paths.

Also, define $v''(P) = \alpha^{-1}v'(P)$ to undo the scaling step without respect to the applied rounding. Since it immediately follows that $v(P) - v''(P) \leq \alpha^{-1}$ for any P we can compare total revenue of the original optimal solution p^* and an optimal solution p'' under the rounded valuations and get that

$$r(p^*) - r(p'') \leq nm\alpha^{-1} = \varepsilon \cdot B \leq \varepsilon \cdot r(p^*).$$

This is due to the fact that we can obtain a solution under the rounded valuations by taking the original optimal solution and reducing the price of each edge by α^{-1}. In this solution, all paths that give any revenue in the optimal solution will still do so. On the other hand, each of the n paths contains at most m edges, bounding our loss on a single path by $m \cdot \alpha^{-1}$. For polynomial running time observe that after the scaling step no declaration with value higher than nm/ε exists.

Theorem 6.3.4. *For any instance of G-SMP with nested paths the FPTAS described above achieves approximation ratio $(1 - \varepsilon)$ in time $O(n^5 m^2 \varepsilon^{-2})$.*

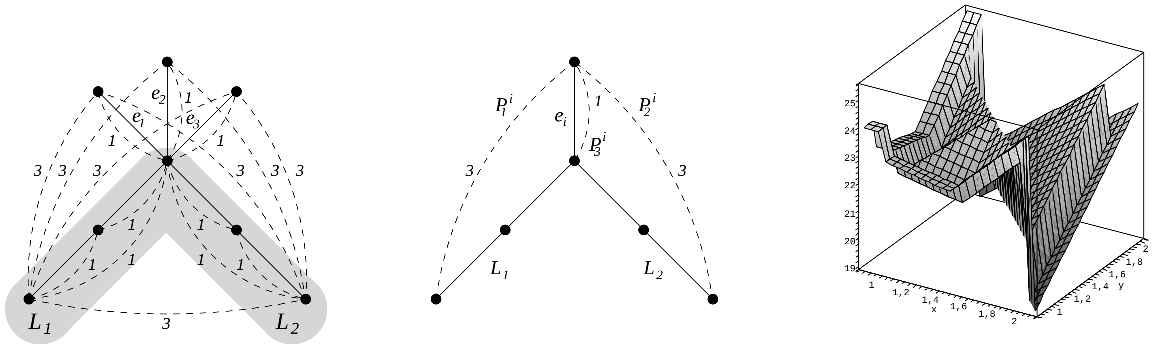
6.4 The Tollbooth Problem

Guruswami et al. [GHK+05] prove that SMP is APX-hard. However, their reduction creates a problem instance in which some products are contained in a constant fraction of all sets. From a technical standpoint, this appears quite unavoidable, since an approximation preserving reduction to SMP always brings up the problem that we need to force optimal (or approximately optimal) solutions to be in some sense well behaved (i.e., close to integral) in order to be able to reconstruct solutions to the (combinatorial) problem that was our reduction's starting point. On the other hand, it is certainly desirable to have hardness results also for sparse instances, especially because it turns out that the number of requests per product is one of the most crucial parameters when it comes to finding good approximations using upper bounding techniques known so far. In what follows we give a proof of inapproximability for sparse instances of G-SMP. We prove APX-hardness by a reduction from MAX-2SAT(3), of which we give a detailed outline below. The technical details of the construction are found in Section 6.4.1.

Remember that the MAX-2SAT problem is defined by a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$ and a collection of m disjunctive clauses of at most 2 literals, where each literal is a variable or negated variable from $\mathcal{V}$. We want to find a truth assignment $t : \mathcal{V} \to \{0, 1\}$ that maximizes the number of satisfied clauses. MAX-2SAT(3) is the special case in which the number of occurrences of each literal is bounded by 3. MAX-2SAT(3) is known to be APX-hard [ACG+99].

We make the simplifying assumption that all clauses have length exactly 2. This is w.l.o.g., since we can replace clauses that consist of only a single literal l by clauses of type $(l \vee y) \wedge (\overline{y} \vee \overline{y})$, where y is a newly added variable. A simple calculation shows that APX-hardness is preserved by this modification.

For each appearance of a literal in the MAX-2SAT(3) instance we define a literal gadget as found in Figure 6.2(a). Literal gadgets are similar to the weight gadgets from the previous section with weight being fixed as 1. As before we will w.l.o.g. not assign prices to the individual edges of a literal gadget but only to the gadget itself, assuming that the price is split evenly among the edges. We note that by Fact 6.3.2 a literal gadget L gives maximum revenue 2 under price assignment p if and only if $p(L) = 1$ or $p(L) = 2$. If two literal occurrences belong to the same clause, their corresponding literal gadgets are combined into a clause gadget as depicted in Figure 6.2(a). The following lemma states that clause gadgets essentially model the behavior of clauses in the SAT instance.



(a) (b) (c)

Figure 6.2: (a) Two literal gadgets L_1 and L_2 are combined into a clause gadget C. (b) In addition to the literal gadgets C contains three copies of the above substructure. (c) Maximum revenue from a clause gadget as a function of $p(L_1), p(L_2) \in [1, 2]$.

Lemma 6.4.1. *Let C be a clause gadget with literal gadgets L_1 and L_2 and assume that $p(L_1), p(L_2) \in [1, 2]$. The maximum revenue obtainable from C is 25. Profit 25 is obtained under price assignment p if and only if $p(L_i) = 2$ and $p(L_j) = 1$, $\{i, j\} = \{1, 2\}$, or $p(L_1) = p(L_2) = 2$. C gives revenue 24 if $p(L_1) = p(L_2) = 1$.*

Lemma 6.4.1 follows directly from the following observation. Once the prices of literal gadgets L_1 and L_2 are fixed, this determines the revenue maximizing prices for edges e_1, e_2 and e_3 (see Fig. 6.2(a)). This allows us to express the optimal revenue from clause gadget C as a piecewise linear function of $p(L_1)$ and $P(L_2)$. This function is depicted in Fig. 6.2(c), details are found in Section 6.4.1. Why we are only interested in the case that $p(L_1), p(L_2) \in [1, 2]$ will become clear in a moment.

We will now define the complete G-SMP instance for our reduction. For each variable x_i the instance will contain 3 literal gadgets corresponding to occurences of literal x_i and 3 literal gadgets corresponding to occurences of literal $\overline{x}_i$, which we denote as $L_0(x_i), L_1(x_i), L_2(x_i)$ and $L_0(\overline{x}_i), L_1(\overline{x}_i), L_2(\overline{x}_i)$. For every clause $c_i = (x_j \vee x_k)$ we construct a clause gadget C_i on literal gadgets $L_h(x_j)$ and $L_l(x_k)$, $h, l \in \{0, 1, 2\}$. Literal gadgets that are not part of any clause gadget, because the corresponding literal occurs less than 3 times, are referred to as *dummy literal gadgets*. A connected graph is obtained by merging the literal gadgets belonging to the same variable into a cycle, i.e., for each variable x_i we connect $L_1(x_i)$ with $L_1(\overline{x}_i)$, then $L_1(\overline{x}_i)$ with $L_2(x_i)$ and so on, until we close the cycle by connecting $L_3(\overline{x}_i)$ with $L_1(x_i)$. Connecting literal gadgets is done by merging their respective start- and end-vertices, where by convention a literal gadget is connected to its clause gadget via its start-vertex. For dummy literal gadgets, this decision is arbitrary. Finally, 6 *literal exclusion paths* $P_1^i, \ldots, P_6^i$ for each variable x_i are defined. Every literal exclusion path P contains the 4 edges belonging to two consecutive literal gadgets that were just joined together and has value $v(P) = 3$. For an illustration of this part of the construction see Fig. 6.3.

A price assignment p on the resulting graph is said to be *integral* if $p(L) \in \{1, 2\}$ for all literal gadgets L. We say that an integral price assignment is SAT-*feasible* if $p(L_0(x_i)) = p(L_1(x_i)) = p(L_2(x_i))$,

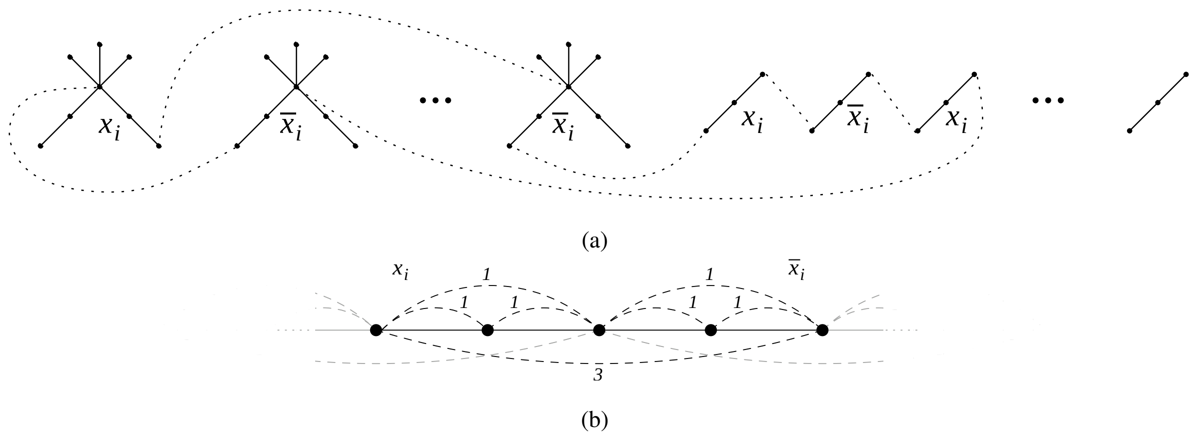


Figure 6.3: (a) Literal gadgets belonging to literals x_i, $\overline{x}_i$ are joined together by merging vertices. (b) A literal exclusion path spans each adjacent pair of literal gadgets.

$p(L_0(\overline{x}_i)) = p(L_1(\overline{x}_i)) = p(L_2(\overline{x}_i))$ and $p(L_0(x_i)) \neq p(L_0(\overline{x}_i))$ for all i. The intuition behind SAT-*feasibility* is quite obvious. We can associate a SAT-feasible price assignment p with a truth assignment t for the original MAX-2SAT(3) instance by setting $t(x_i) = 1$ if $p(L_0(x_i)) = 2$ and $t(x_i) = 0$ else. On the constructed graph SAT-feasible prices result in maximum revenue from all dummy literal gadgets and literal exclusion paths. Revenue from each clause gadget is 24 or 25 depending on whether at least one of the contained literal gadgets has price 2. Thus, total revenue of price assignment p is then directly related to the number of clauses satisfied by t.

The main step towards our hardness result lies in proving that we can transform in polynomial time an arbitrary price assignment p on the constructed graph into a SAT-feasible price assignment p^* of no smaller revenue. This also yields that the optimal price assignment can be assumed to be SAT-feasible and, thus, gives us an easy way of upper bounding the optimal revenue obtainable on the constructed G-SMP instance. Given any price assignment p, Algorithm 4 returns a SAT-feasible price assignment p^*.

Lemma 6.4.2. *For price assignments p and p^* as defined by Algorithm 4 it holds that $r(p^*) \geq r(p)$.*

The rather lengthy proof of Lemma 6.4.2 is found in Section 6.4.1. For a given MAX-2SAT(3) instance with m clauses over n variables the optimal truth assignment can immediately be turned into an optimal solution for our G-SMP instance with revenue $18n+24m+2d+43v+\vartheta^*$, where ϑ^* refers to the maximum number of satisfiable clauses in the formula, v is the number of variables added due to clauses of length 1 and d denotes the number of dummy literal gadgets in the graph. We note that $\vartheta^* \geq m/2$ and, thus, the expression $18n + 24m + 2d + 43v$ can be upper bounded by $246\vartheta^*$ using trivial bounds on n, d and v. Assume now we had an algorithm with approximation guarantee $1 - \varepsilon$ for G-SMP. We can write the revenue of any price assignment returned by the algorithm as $18n + 24m + 2d + 43v + \vartheta$ and construct a corresponding truth assignment satisfying ϑ clauses. By our assumption

$$1 - \varepsilon \leq \frac{18n + 24m + 2d + 43v + \vartheta}{18n + 24m + 2d + 43v + \vartheta^*} \leq \frac{246\vartheta^* + \vartheta}{247\vartheta^*},$$

and, thus, $\vartheta/\vartheta^* \geq 1 - 247\varepsilon$. Choosing constant ε sufficiently small yields constant approximation guarantees arbitrarily close to 1 for MAX-2SAT(3).

Algorithm 4: Transforming prices p into SAT-feasible prices p^*.

// Step 1: $p \to p'$
for *all literal gadgets* L **do**
  if $p(L) \notin [1,2]$ **then**
    Set $p'(L) = 1$ if $p(L) < 1$, set $p'(L) = 2$ if $p(L) > 2$.
  else
    Set $p'(L) = p(L)$.

// Step 2: $p' \to p''$
Let $p'' = p'$.
for *all literal exclusion paths* P **do**
  Let P be connecting literal gadgets L_1, L_2 and assume $p''(L_1) \leq p''(L_2)$.
  if $p''(L_1) + p''(L_2) > 3$ **then**
    Set $p''(L_1) = 1$.

// Step 3: $p'' \to p^*$
Choose p^* as the SAT-feasible price assignment that minimizes

$$|\{L \mid p''(L) > 1.75 \wedge p^*(L) = 1\}|.$$

Theorem 6.4.3. *G-SMP is* APX*-hard on instances where*

- *the length of each path is bounded by a constant $\ell \geq 4$,*
- *the valuations for all paths are in $\{1,2,3\}$,*
- *there is at most a single offer for each possible path,*
- *the number of paths in which each edge is contained is bounded by a constant $B \geq 8$ and*
- *the underlying graph's maximum degree is bounded by a constant $\Delta \geq 7$.*

Considering the underlying graph structure, the above APX-hardness result is in a sense best possible. Guruswami et al. [GHK$^+$05] present a pseudopolynomial time algorithm for the highway problem with paths of constant length. This approach also yields an FPTAS and can in fact easily be generalized to the case of constant degree trees instead of a line. Hence, APX-hardness is lost if we simultaneously require a tree and a maximum degree bounded by a constant.

6.4.1 Full Proof of Theorem 6.4.3

Below we give all the technical details from the preceding proof of APX-hardness. Lemma 6.4.4 gives a precise characterization of the revenue obtainable from a clause gadget depending on the prices assigned to its literal gadgets. This immediately yields Lemma 6.4.1. Lemmas 6.4.5, 6.4.6 and 6.4.7 prove that Algorithm 4 does not decrease the revenue obtained by the initial price assignment.

Proof of Lemma 6.4.1

Lemma 6.4.4 shows how the revenue obtained from a clause gadget can be expressed as a piecewise linear function that depends solely on the prices assigned to the contained literal gadgets L_1 and L_2, if these are within a reasonable range (which is ensured by Step 1 of Algorithm 4). This is due to the fact that the additional edges e_1, e_2 and e_3 (see Fig. 6.2) are contained only in paths that are part of this gadget and, thus, their optimal prices are determined by $p(L_1)$ and $p(L_2)$. Lemma 6.4.1 then follows as a simple corollary.

Lemma 6.4.4. *Let clause gadget C consist of literal gadgets L_1, L_2. For $p(L_1), p(L_2) \in [1, 2]$ we can describe the revenue $r(C)$ obtained from clause gadget C as a function of $p(L_1)$ and $p(L_2)$ as follows:*

$$r(C) = \begin{cases} (1)\ 24, & \text{if } p(L_1) = p(L_2) = 1 \\ (2)\ 24 - p(L_2), & \text{if } 1 = p(L_1) < p(L_2) \leq \frac{3}{2} \\ (3)\ 15 + 5p(L_2), & \text{if } 1 = p(L_1), \frac{3}{2} < p(L_2) \\ (4)\ 18 + 5p(L_1) - p(L_2), & \text{if } 1 < p(L_1) \leq p(L_2) \leq \frac{3}{2} \\ (5)\ 9 + 5p(L_1) + 5p(L_2), & \text{if } 1 < p(L_1) < \frac{3}{2} < p(L_2) \text{ and } p(L_1) + p(L_2) \leq 3 \\ (6) - (9) \text{ symmetric to (2) - (5) with } p(L_1) > p(L_2) & \\ (10)\ 9 + 4p(L_1) + 4p(L_2), & \text{if } p(L_1) + p(L_2) > 3 \end{cases}$$

Proof: A clause gadget consists of its literal gadgets and three copies of the substructure found in Figure 6.2(b). In order to prove the claim we just need to describe how $p(e_i)$ has to be chosen to maximize total revenue. Consider the substructure in Figure 6.2(b) consisting of edge e_i and the paths defined on e_i, L_1 and L_2. For any given $p(L_1) \leq p(L_2)$ in $[1, 2]$ there are only two possible prices that can potentially give maximum revenue. These are $p(e_i) = 1$ or $p(e_i) = 3 - p(L_2)$, respectively. To see this, note, that for any $p(e_i) \leq 1$ edge e_i will contribute $3p(e_i)$ to the substructure's total revenue ($p(e_i)$ for each path in which it appears). It follows that any price assignment p with $p(e_i) < 1$ can be improved by setting $p(e_i) = 1$. Now consider p with $p(e_i) > 1$. For $p(e_i) \leq 3 - p(L_2)$ edge e_i contributes revenue $2p(e_i)$ (being counted on two paths). Thus, any price assignment p with $1 < p(e_i) < 3 - p(L_2)$ can be improved by setting $p(e_i) = 3 - p(L_2)$. Especially, since $p(L_2) \leq 2$, it follows that $2p(e_i) \geq 2$. For $p(e_i) > 3 - p(L_2)$ the contribution becomes $p(e_i)$, since the price for the path containing L_2 exceeds its threshold and edge e_i is counted only once on the path containing L_1. From $p(L_1) \geq 1$ it follows that e_i can contribute at most 2 on this path and, thus, revenue does not decrease by setting $p(e_i) = 3 - p(L_2)$.

Using this observation it is clear how $p(e_i)$ must be chosen in each case with $p(L_1) \leq p(L_2)$. We let $p(e_i) = 3 - p(L_2)$ whenever $2(3 - p(L_2)) > 3$ (or, equivalently, $p(L_2) < 3/2$) and $p(e_i) = 1$ otherwise. Cases with $p(L_1) > p(L_2)$ are symmetric, thus, $p(e_i) = 1$ or $p(e_i) = 3 - p(L_1)$. Total revenue from clause gadget C is then obtained by summing up over 3 copies of the substructure, 2 literal gadgets and path P. □

Proof of Lemma 6.4.2

Algorithm 4 in Section 6.4 transform an arbitrary price assignment p into a SAT-feasible price assignment p^*. Let p' and p'' denote the intermediate prices as defined in Algorithm 4. Lemmas 6.4.5, 6.4.6 and 6.4.7 show that the overall revenue does not decrease in any step of the algorithm.

Lemma 6.4.5. *It holds that* $r(p') \geq r(p)$.

Proof: We show that revenue does not decrease on any literal exclusion path, in any clause or dummy literal gadget. For dummy literal gadgets this is trivial, since revenue from these gadgets becomes maximal if the assigned price is changed. Let us then fix some clause gadget C with literal gadgets L_1 and L_2. We assume w.l.o.g. that $p(L_1) \leq p(L_2)$ and consider all possible cases. By $r(C)$ and $r'(C)$ we refer to the total revenue under p and p', respectively.

Case (1) $p(L_1), p(L_2) < 1 \Rightarrow p'(L_1) = p'(L_2) = 1$. The revenue from each substructure (Fig. 6.2(b)) is bounded by 6, thus,

$$r(C) \leq 3 \cdot 6 + 3(p(L_1) + p(L_2)) \leq 24 = r'(C),$$

by summing up over 3 substructures, 2 literal gadgets and the connecting path P.

Case (2) $p(L_1) < 1 \leq p(L_2) \leq 2 \Rightarrow p'(L_1) = 1, p'(L_2) = p(L_2)$. From the proof of Lemma 6.4.4 we know that maximum revenue from each substructure is obtained by setting $p(e_i) = 1$ or $p(e_i) = 3 - p(L_2)$ depending on $p(L_2)$. In both cases increasing $p(L_1)$ to 1 gives an increase in revenue of $1 - p(L_1)$ in each substructure. Identical observations are easily made for the literal gadgets and path P. It follows that

$$r(C) \leq r(C) + 6(1 - p(L_1)) = r'(C).$$

Case (3) $p(L_1) < 1, p(L_2) > 2 \Rightarrow p'(L_1) = 1, p'(L_2) = 2$. By Lemma 6.4.4 revenue from C is maximal under p' and the claim follows.

Case (4) $1 \leq p(L_1) \leq 2 < p(L_2) \Rightarrow p'(L_1) = p(L_1), p'(L_2) = 2$. From observations analogous to those in the proof of Lemma 6.4.4 it follows that under price assignment p it must be $p(e_i) = 3 - p(L_2)$ in order to obtain maximum revenue $3 + 2(3 - p(L_2)) + p(L_1)$ from each substructure. (If $p(L_2) > 3$ we let $p(e_i) = 0$ and the former is an upper bound on the revenue from each substructure.) Under p' setting $p'(e_i) = 1$ leads to revenue $5 + p(L_1)$. With $3 - p(L_2) < 1$ we can conclude that revenue increases in each substructure. Literal gadget L_2 and path P give revenue 0 under p and, thus, cannot contribute less under p'.

Case (5) $p(L_1), p(L_2) > 2 \Rightarrow p'(L_1) = p'(L_2) = 2$. By Lemma 6.4.4 revenue from C is maximal under p' and the claim follows.

We then look at a single literal exclusion path P connecting literal gadgets L_1 and L_2. We have to distinguish the following 2 cases.

Case (a) $p(L_1) + p(L_2) \leq 3$. We observe that $p(L_1) + p(L_2) \leq p'(L_1) + p'(L_2) \leq 3$ and, since in this case revenue from P is just the sum of these prices, it follows that revenue under p' is no smaller than under p.

Case (b) $p(L_1) + p(L_2) > 3$. Then the revenue from P under price assignment p is 0 and can obviously only increase when going to p'.

Thus, we have shown that $r'(C) \geq r(C)$. □

Lemma 6.4.6. *It holds that* $r(p'') \geq r(p')$.

Proof: Price assignment p'' is constructed by iterating over all literal exclusion paths and modifying p' locally. We will consider a single iteration and prove that revenue can only be increased by the modifications.

Let P be literal exclusion path connecting literal gadgets L_1 and L_2 and look at the iteration in which P is considered. If $p'(L_1) + p'(L_2) \leq 3$ then nothing is changed and obviously revenue remains the same. Assume then that $p'(L_1) + p'(L_2) > 3$. We let w.l.o.g $p'(L_1) \leq p'(L_2)$ and, thus, will have $p''(L_1) = 1, p''(L_2) = p'(L_2)$. The revenue under p' and p'' may differ in the following 3 places: the clause gadget C in which literal gadget L_1 is contained, literal exclusion path P and the second literal exclusion path Q in which L_1 is contained. We refer to the change of revenue in these places as Δ_C, Δ_P, Δ_Q and note that the total change of revenue caused by changing $p'(L_1)$ can be written as $\Delta = \Delta_C + \Delta_P + \Delta_Q$. We will bound each summand individually.

Consider clause gadget C that consists of L_1 and some other literal gadget L_3. We go through all possible cases and apply Lemma 6.4.4. Note, that $p'(L_1) > 1$.

Case (1) $1 = p'(L_3) < p'(L_1) \leq 3/2$. Due to the changed price we jump from Case (2) to Case (1) in Lemma 6.4.4. Hence, $\Delta_C \geq 24 - (24 - p'(L_1)) \geq 0$.

Case (2) $p'(L_3) = 1, p'(L_1) > 3/2$. We jump from (3) to (1), thus, $\Delta_C \geq 24 - (15 + 5p'(L_1)) \geq -1$.

Case (3) $1 < p'(L_3) \leq p'(L_1) \leq 3/2$. We jump from Case (4) to Case (6), thus,

$$\begin{aligned}\Delta_C &= (24 - p'(L_3)) - (18 + 5p'(L_3) - p'(L_1)) \\ &= 6 - 5p'(L_3) \geq -3/2.\end{aligned}$$

Case (4) $1 < p'(L_3) < 3/2 < p'(L_1), p'(L_1) + p'(L_3) \leq 3$. We jump from Case (5) to Case (6), thus,

$$\begin{aligned}\Delta_C &= (24 - p'(L_3)) - (9 + 5p'(L_3) + 5p'(L_1)) \\ &= 15 - 5(p'(L_1) + p'(L_3)) - p'(L_3) \geq -3/2.\end{aligned}$$

Case (5) $1 < p'(L_1) \leq p'(L_3) \leq 3/2$. We jump from Case (8) to case (6), thus,

$$\begin{aligned}\Delta_C &= (24 - p'(L_3)) - (18 + 5p'(L_1) - p'(L_3)) \\ &= 6 - 5p'(L_1) \geq -3/2.\end{aligned}$$

Case (6) $1 < p'(L_1) < 3/2 < p'(L_3), p'(L_1) + p'(L_3) \leq 3$. We jump from Case (9) to Case (7), thus,

$$\begin{aligned}\Delta_C &= (15 + 5p'(L_3)) - (9 + 5p'(L_1) + 5p'(L_3)) \\ &= 6 - 5p'(L_1) \geq -3/2.\end{aligned}$$

Case (7) $p'(L_1) + p'(L_3) > 3$. If $p'(L_3) \leq 3/2$ then we jump from Case (10) to Case (6) and have

$$\begin{aligned}\Delta_C &= (24 - p'(L_3)) - (9 + 4p'(L_1) + 4p'(L_3)) \\ &= 15 - 4(p'(L_1) + p'(L_3)) - p'(L_3) \\ &\geq 15 - 4(2 + 3/2) - 3/2 \geq -1/2.\end{aligned}$$

If $p'(L_3) > 3/2$ then we jump from Case (10) to Case (7) and have

$$\begin{aligned} \Delta_C &= (15 + 5p'(L_3)) - (9 + 4p'(L_1) + 4p'(L_3)) \\ &= 6 - 4p'(L_1) + p'(L_3) \geq 6 - 8 + 3/2 \geq -1/2. \end{aligned}$$

We conclude that $\Delta_C \geq -3/2$. Consider then literal exclusion path P. Since $p'(L_1) + p'(L_2) > 3$ it follows that P gives revenue 0 under price assignment p'. From $p'(L_1) \leq p'(L_2)$ we conclude that $p'(L_2) \geq 3/2$. With $p''(L_1) = 1$ path P then gives revenue at least $5/2$ under p'' and we have that $\Delta_P \geq 5/2$. It is $p'(L_1) - p''(L_1) \leq 1$. Since the revenue on literal exclusion path Q can decrease by no more than this difference we observe $\Delta_Q \geq -1$. We can then bound the total difference in revenue by

$$\Delta = \Delta_C + \Delta_P + \Delta_Q \geq -3/2 + 5/2 - 1 = 0,$$

which finishes the proof. □

Lemma 6.4.7. *It holds that* $r(p^*) \geq r(p'')$.

Proof: For each variable of the MAX-2SAT(3) instance our G-SMP instance contains 6 corresponding literal gadgets, which form a cyclic structure as depicted in Fig. 6.3(a). On each such cycle we have 6 literal exclusion paths as found in Fig. 6.3(b).

We first note that the SAT-feasible price assignment p^* can be constructed locally, i.e., considering only the literal exclusion paths belonging to one variable at a time. We will also follow this local approach to show that total revenue does not decrease. For variable x_i we define $\mathcal{X}_i = \{L_0(x_i), L_1(x_i), L_2(x_i)\}$, $\overline{\mathcal{X}}_i = \{L_0(\overline{x}_i), L_1(\overline{x}_i), L_2(\overline{x}_i)\}$ and let

$$\alpha_i = |\{L \mid L \in (\mathcal{X}_i \cup \overline{\mathcal{X}}_i) \wedge p''(L) > 1.75 \wedge p^*(L) = 1\}|$$

denote the number of *problematic* literal gadgets belonging to variable x_i. We start by observing that $\alpha_i \in \{0, 1\}$. Let $B = \{L \mid L \in (\mathcal{X}_i \cup \overline{\mathcal{X}}_i) \wedge p''(L) > 1.75\}$. If $B \subseteq \mathcal{X}_i$ or $B \subseteq \overline{\mathcal{X}}_i$ then we can obviously define a SAT-feasible p^* such that $\alpha_i = 0$. So let us assume that $B \cap \mathcal{X}_i \neq \emptyset$ and $B \cap \overline{\mathcal{X}}_i \neq \emptyset$. From the construction of p'' we know that B cannot contain 2 literal gadgets that are connected by a literal exclusion path, since one of the prices would have been set to 1 in Step 2 of the transformation. From the fact that we are looking at a cyclic structure of length 6 it then follows that $|B| = 2$ and we can assume w.l.o.g that $B = \{L_0(x_i), L_1(\overline{x}_i)\}$. It is then clear that any SAT-feasible price assignment results in $\alpha_i = 1$.

Let Δ_i denote the change in revenue in all literal exclusion paths belonging to variable x_i going from price assignment p'' to p^*. We will now show that $\Delta_i \geq \alpha_i$. To see this, first note that $\Delta_i \geq 0$ is a trivial observation, since revenue from the literal exclusion paths becomes maximal under p^*. It then only remains to be shown that $\Delta_i \geq 1$ if $\alpha_i = 1$. We have already argued that we can assume w.l.o.g that $B = \{L_0(x_i), L_1(\overline{x}_i)\}$. Again from the construction of p'' it follows that $p''(L_1(x_i))$, $p''(L_2(x_i))$, $p''(L_0(\overline{x}_i))$ and $p''(L_2(\overline{x}_i))$ are bounded above by 1.25, since they are connected with one of the literal gadgets from B through a literal exclusion path. Profit from paths P_2^i and P_5^i under p'' is then bounded by 2.5 each. Hence, total revenue from the cycle is at most $4 \cdot 3 + 2 \cdot 2.5 = 17$ under p'' and will increase to its maximum of 18 under p^*, thus, $\Delta_i \geq 1$.

For the second part of the proof we now consider an arbitrary clause gadget C consisting of some literal gadgets L_1, L_2 and let Δ_C refer to the relative change of revenue. Under price assignment p^* each clause gadget gives at least revenue 24. From Lemma 6.4.4 it follows that revenue from C under p'' is at most 24 if $p''(L_1), p''(L_2) \leq 1.75$. We note that $\Delta_C \geq 0$ in this case. But how can Δ_C become negative? This can happen only if there is a $j \in \{1, 2\}$ such that $p''(L_j) > 1.75$ and $p^*(L_1) = p^*(L_2) = 1$. It is clear that $\Delta_C \geq -1$ in this case, since revenue from a clause gadget is at least 24 in any SAT-feasible price assignment.

The important observation now is that each variable i with $\alpha_i = 1$ can cause at most one clause gadget to have decreasing revenue. It immediately follows that $\sum_C \Delta_C \geq -\sum_i \alpha_i$. Finally, we note that revenue from any dummy literal gadget is maximal under p^* and, thus, can only be higher than under p''. Now let Δ be the change of revenue in the complete G-SMP instance. From the above argumentation it is immediately clear that

$$\Delta \geq \sum_{i=1}^{n} \Delta_i + \sum_{C} \Delta_C \geq \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i = 0.$$

This finishes the proof. □

6.5 An $\mathcal{O}(\log \ell + \log B)$-Approximation

In [DFHS06] it is shown that the single-price algorithm cannot be beaten on general SMP. In light of our results in the previous sections of this chapter it is apparent that the problem remains difficult to approximate even in very restrictive cases, yet, it should be possible to design algorithms that are capable of exploiting the special structure of sparse problem instances to obtain constant approximation guarantees at least in some cases.

We consider SMP in general, i.e., without the assumption of an underlying network, and focus on cases in which both the maximum size ℓ of any set and the maximum number B of different sets in which any single product appears are bounded better than trivially. We present an algorithm with approximation guarantee $\mathcal{O}(\log \ell + \log B)$, which asymptotically matches the single-price algorithm in the worst case, since obviously $\ell \leq n$ and $B \leq m$, but achieves improved ratios on sparse problem instances.

Remember that $\delta(S)$ denotes the price per item of set S. Algorithm 5 is based on the idea of partitioning the sets according to their δ-values.

The crucial step in the analysis of Algorithm 5 consists of showing that we do not lose too much potential revenue by removing sets in Step 7. This claim is formalized in the following lemma.

Lemma 6.5.1. *Let $\mathcal{S}_i$ and $\mathcal{S}_i^*$ be defined as in Algorithm 5. Then* $\sum_{S \in \mathcal{S}_i^*} |S| \delta'(S) \geq \sum_{S \in \mathcal{S}_i \setminus \mathcal{S}_i^*} |S| \delta'(S)$.

Proof: A set S is removed from $\mathcal{S}_i^*$ if it intersects with another set S' whose δ'-value is maximal among the sets that are still in $\mathcal{S}_i^*$. Looking at this the other way around, it means that a set that causes other sets to be removed from $\mathcal{S}_i^*$ is never removed itself.

Algorithm 5: An $\mathcal{O}(\log \ell + \log B)$-approximation for SMP.

Let $\delta'(S) = 2^{\lfloor \log \delta(S) \rfloor}$ for all $S \in \mathcal{S}$.
Partition $\mathcal{S}$ into $\mathcal{S}_0, \ldots, \mathcal{S}_t$, where

$$\mathcal{S}_i = \left\{ S \in \mathcal{S} \mid \delta'(S) \in \left\{ 2^{i+j \cdot \lceil \log(\ell^2 B) \rceil} \mid j \in \mathbb{N}_0 \right\} \right\}$$

and $t = \lceil \log(\ell^2 B) \rceil - 1$.
for $i = 0, \ldots, t$ **do**
    Let $\mathcal{S}_i^* = \mathcal{S}_i$.
    while *$\mathcal{S}_i^*$ contains sets S, S' with $S \cap S' \neq \emptyset$ and $\delta'(S) \neq \delta'(S')$* **do**
        Let S be such a set with maximum value $\delta'(S)$.
        Remove all S' with $S \cap S' \neq \emptyset$ and $\delta'(S') < \delta'(S)$ from $\mathcal{S}_i^*$.
    Define prices p_i as $p_i(u) = \delta'(S)$ if u is still contained in a set $S \in \mathcal{S}_i^*$ and $p_i(u) = 0$ else.
Return $p = \operatorname{argmax}\{r(p_i) \mid p_i, i = 0, \ldots, t\}$.

For a set $S \in \mathcal{S}_i^*$ let then $R(S) \subset \mathcal{S}_i$ denote the sets that are removed due to their intersection with S. Clearly, for every $S' \in R(S)$ it must be true that

$$\delta'(S') \leq (\ell^2 B)^{-1} \cdot \delta'(S),$$

by our partitioning. On the other hand, S contains at most ℓ different elements, each of which is contained in at most $B-1$ further sets, thus, $|R(S)| \leq \ell(B-1)$. It follows that

$$\begin{aligned} \sum_{S \in \mathcal{S}_i \setminus \mathcal{S}_i^*} |S|\delta'(S) &= \sum_{S \in \mathcal{S}_i^*} \sum_{S' \in R(S)} |S'|\delta'(S') \\ &\leq \sum_{S \in \mathcal{S}_i^*} \ell(B-1)\ell(\ell^2 B)^{-1}\delta'(S) \\ &\leq \sum_{S \in \mathcal{S}_i^*} |S|\delta'(S), \end{aligned}$$

which proves the claim. □

Applying Lemma 6.5.1 immediately yields the desired approximation guarantee. We simply argue that sets are partitioned into $\mathcal{O}(\log \ell + \log B)$ many classes and use the sum of values of the sets in any single class as an upper bound on the maximum revenue obtainable from that class.

Theorem 6.5.2. *Algorithm 5 computes an $\mathcal{O}(\log \ell + \log B)$-approximation to* SMP *with sets of size at most ℓ and at most B requests per product.*

Proof: We observe that $\delta'(S) = \delta'(T)$ for any $S, T \in \mathcal{S}_i^*$ with $S \cap T \neq \emptyset$. Therefore, it holds that $r(p_i) = \sum_{S \in \mathcal{S}_i^*} |S|\delta'(S)$ for $0 \leq i \leq t$. Choosing $p = \operatorname{argmax}\{r(p_i) \mid p_i, i = 0, \ldots, t\}$ and applying

Lemma 6.5.1 yields

$$\begin{aligned} r(p) &\geq \frac{1}{t+1}\sum_{i=0}^{t}\sum_{S\in\mathcal{S}_i^*}|S|\delta'(S) \\ &\geq \frac{1}{2(t+1)}\sum_{i=0}^{t}\sum_{S\in\mathcal{S}_i}|S|\delta'(S) \\ &= \frac{1}{2(t+1)}\sum_{S\in\mathcal{S}}|S|\delta'(S) \\ &\geq \frac{1}{4(t+1)}\sum_{S\in\mathcal{S}}|S|\delta(S) \geq \frac{1}{4(t+1)}r(p^*), \end{aligned}$$

and, since $t = \mathcal{O}(\log \ell + \log B)$, the claim follows. □

Similar to our argumentation in the proof of Theorem 2.2.1 it is a straightforward observation that using the sum of values of all sets as an upper bound on the optimal revenue cannot result in an approximation guarantee better than $\mathcal{O}(\log B)$. As a tight example consider a single good u and sets $S_1, \ldots, S_B$ with $v(S_i) = 1/i$ and $S_i = \{u\}$ for all i. Any price $p(u) = 1/i$ results in revenue 1 while our upper bounding technique yields $H_B = \Omega(\log B)$.

6.6 Literature

The single-minded pricing problem was first considered by Guruswami et al. [GHK+05], who prove that the single-price algorithm achieves approximation guarantee $H_m + H_n$ and prove APX-hardness of general G-SMP. For the special case of G-SMP in rooted trees where all paths start at the root they provide a polynomial time algorithm. Considering the highway problem, they derive FPTAS's for the cases that all paths are of constant length or all valuations are constant.

Demaine et al. [DFHS06] provide a matching lower bound for the single-price algorithm on general SMP, which was in fact the first known super-constant lower bound for any combinatorial pricing problem.

Balcan and Blum [BB06] improve the approximation guarantee for the general highway problem to $\mathcal{O}(\log n)$. Their algorithm works by partitioning the paths in a clever way and also yields a constant approximation guarantee if path lengths vary by no more than a constant factor. Furthermore, they present the random-partitioning algorithm (Algorithm 2) achieving approximation guarantee $\mathcal{O}(\ell)$ for SMP with sets of bounded size.

The current state-of-the-art algorithm for the highway problem is found in [ESZ07], where Elbassioni et al. present a quasi-polynomial time PTAS using a quite involved dynamic programming approach originally proposed by Bansal et al. [BCES06] for the unsplittable flow problem on line graphs.

A number of other special cases of SMP have also received attention. Grigoriev et al. [GvLS+07] investigate SMP with *comparable items*, i.e., under the restriction that the price of any bundle of products should be monotone in the size of the bundle. They derive a PTAS for this case and prove strong NP-hardness. Hartline and Koltun [HK05] present FPTAS's for SMP with a constant number of distinct products. Balcan and Blum [BB06] consider the *graph vertex pricing problem*, in which sets are of size 1 or 2 and, thus, we can think of consumers as edges in a graph (with self-loops) willing to purchase the incident vertices subject to some budget constraint. They present a 4-approximation for this problem. Krauthgamer et al. [KMR07] consider the special case of graph vertex pricing in which all consumers have the same budget and present a 1.15-approximation based on their work on unit-demand pricing in bimodal markets.

Some work on mechanism design in the single-minded setting is found, e.g., in [AT01], [MN02], or [BKV05]. The results from this chapter have been published in [BK06].

7 Network Pricing II: Stackelberg Games

Our approach to network pricing in Chapter 6 was guided by the idea that consumers behave single-minded, i.e., consider only a single path to connect their terminals. While single-mindedness is sometimes arguably a realistic assumption in general multi-product pricing, the network setting immediately brings up the question why consumers should restrict themselves in this way. Thinking of a public transportation network, passengers might be interested only in connections that are not significantly longer than their shortest alternative, but they would certainly be willing to settle for something slightly longer if this promises to save them considerable money in return. In other words, one would expect consumer preferences to be somewhat more complex than the cases we have discussed so far.

The modelling of consumer preferences and corresponding protocols allowing consumers to specify what they are interested in has received considerable attention in the context of *algorithmic mechanism design* [NR99] and *combinatorial auctions* [CSS06]. The established models range from relatively simple bidding languages to bidders that are represented by oracles allowing certain types of queries, e.g., revealing the desired bundle of items given some fixed set of prices. The latter would be a somewhat problematic assumption in the theory of pricing algorithms, where we usually assume to have access to a rather large number of potential consumers through some sort of sampling procedure and, thus, are interested in preferences that allow for a compact kind of representation.

In this chapter we continue our investigation of network pricing and focus on consumers that have non-trivial preferences, yet can be fully described by their *types* and *budgets* and do not require any kind of oracles. Assume that a company owns a subset of the links in a given network. The remaining edges are owned by other companies and have fixed publicly known prices and some consumer needs to purchase a path between two terminals in the network. Essentially, all feasible paths are the same to her and, since she is acting rational, she is going to buy the cheapest path connecting her terminals. How should we set the prices on the pricable edges in order to maximize the company's revenue? What if there is another consumer, who needs to purchase, e.g., a minimum cost spanning tree?

This type of pricing problem, in which preferences are implicitly defined in terms of some optimization problem, is usually referred to as *Stackelberg pricing* [vS34]. In the standard 2-player form we are given a *leader* setting the prices on a subset of the network edges and a *follower* seeking to purchase a min-cost network satisfying her requirements.

The 2-player shortest path version of this problem has first been considered by Labbé et al. [LMS98]. The first polynomial time algorithm with provable approximation guarantee for this problem is presented by Roch et al. [RSM05]. Their rather involved recursive algorithm yields an approximation ratio that is logarithmic in the number of pricable edges. Bouhtou et al. [BGvH$^+$04] extend the problem to multiple *weighted* followers, i.e., followers that need to route different demands along their paths, and present algorithms for a restricted shortest path problem on parallel links, which they term the *river tarification problem*. For an overview of most of the initial work on Stackelberg network pricing the reader is referred to [vH06].

More recently, Cardinal et al. [CDF$^+$07] investigated the corresponding 2-player minimum spanning tree pricing problem, obtaining a logarithmic approximation guarantee by applying the single-price algorithm and proving that the problem is APX-hard. Their analysis of the single-price algorithm can be extended to show that it achieves similar approximation guarantees for any matroid based Stackelberg game.

Our first result is a novel analysis of the single-price algorithm that proves the same tight approximation guarantee for all Stackelberg pricing problems, independently of the matroid structure. We then consider a special case of 2-player Stackelberg pricing, in which the follower needs to purchase a vertex cover in a given bipartite graph. We show that in contrast to all previously investigated versions, this problem variation allows us to achieve constant approximation guarantees in general and is even polynomial time solvable in a special case. Finally, we take a look at Stackelberg pricing with multiple followers and obtain tight bounds for the unweighted case. For weighted followers, we present a lower bound that resolves an open problem from [BGvH$^+$04] and a novel analysis of the single-price algorithm that yields first evidence that approximation guarantees independent of the number of followers are possible.

The rest of the chapter is organized as follows. A formal problem definition is found in Section 7.1. Sections 7.2 through 7.3 contain our results on the single-price algorithm and the bipartite vertex cover pricing game. The extensions to multiple followers are found in Section 7.4. Section 7.5 reviews some related literature.

7.1 Preliminaries

We can view Stackelberg pricing as a class of multi-player 1-round games. Let $G = (V, E)$ be a multi-graph. There are two types of players in the game, one *leader* and one or more *followers*. We consider two classes of *edge* and *vertex games*, in which either the edges or the vertices have costs. For most of this chapter, we will consider edge games, but the definitions and results for vertex games follow analogously. In an edge game, the edge set E is divided into two sets $E = E_p \cup E_f$ with $E_p \cap E_f = \emptyset$. For the set of *fixed-price* edges E_f there is a fixed cost $c(e) \geq 0$ for each edge $e \in E_f$. For the set of *pricable* edges E_p the leader can specify a price $p(e) \geq 0$ for each edge $e \in E_p$. We denote the number of pricable edges by $m = |E_p|$. Each follower $i = 1, \ldots, k$ has a set $\mathcal{S}_i \subset 2^E$ of *feasible* subnetworks. By $p(S) = \sum_{e \in S \cap E_p} p(e)$ and $c(S) = \sum_{e \in S \cap E_f} c(e)$ we refer to the total cost due to pricable or fixed-cost edges in subnetwork $S \in 2^E$, respectively. The total *weight* of a subnetwork is defined as $w(S) = p(S) + c(S)$.

Throughout the chapter we assume that for any price function p every follower i can in polynomial time find a subnetwork $S_i^*(p)$ of minimum weight. As before, we want to find the price assignment p^* for the leader that maximizes the sum of prices of the edges bought by the followers.

Definition 7.1.1. *Given a graph $G = (V, E_f \cup E_p)$ with fixed costs $c(e)$ on edges $e \in E_f$ and k followers with feasible subnetworks $\mathcal{S}_1, \ldots, \mathcal{S}_k$, the Stackelberg Network Pricing Problem (*STACK*) asks for prices $p : E_p \to \mathbb{R}_0^+$ maximizing*

$$r(p) = \sum_{i=1}^{k} p(S_i^*(p)).$$

We again denote the optimal prices as p^* and let $r(p^*)$ refer to the corresponding maximum revenue. To guarantee that the maximum revenue is bounded and the optimization problem is non-trivial, we assume that there is at least one feasible subnetwork for each follower i that is composed only of fixed-price edges. In order to avoid technicalities, we assume w.l.o.g. that among subnetworks of identical weight the follower always chooses the one with higher revenue for the leader. It is not difficult to see that in the 2-player case we also need followers with a large number of feasible subnetworks in order to make the problem interesting.

Proposition 7.1.2. *Given follower i and a fixed subnetwork $S_i \in \mathcal{S}_i$, we can compute prices p with $w(S_i) = \min_{S \in \mathcal{S}_i} w(S)$ maximizing $p(S_i)$ or decide that such prices do not exist in polynomial time. In the 2-player game, if $|\mathcal{S}| = \mathcal{O}(poly(m))$, revenue maximization can be done in polynomial time.*

Proof: Fix follower i and subnetwork $S_i \in \mathcal{S}_i$. We formulate the problem of extracting maximum revenue from S_i as the following LP, where variable x_e defines the price of edge $e \in E_p$:

$$\text{max.} \quad \sum_{e \in S_i \cap E_p} x_e \tag{7.1}$$

$$\text{s.t.} \quad \sum_{e \in S_i \cap E_p} x_e + \sum_{e \in S_i \cap E_f} c(e) \leq \sum_{e \in S \cap E_p} x_e + \sum_{e \in S \cap E_f} c(e) \quad \forall S \in \mathcal{S}_i \tag{7.2}$$

$$x_e \geq 0 \tag{7.3}$$

Constraints 7.2 require that S_i is the cheapest feasible network for follower i, formally $w(S_i) \leq w(S)$ for all feasible networks $S \in \mathcal{S}_i$. Clearly the number of these constraints might be exponential in m. However, by our assumption we can compute the min-cost subnetwork for any given set of prices and, thus, have a polynomial time separation oracle.

Now assume that $|\mathcal{S}| = \mathcal{O}(poly(m))$ in the 2-player case. By enumerating all $S \in \mathcal{S}$ and optimizing revenue for each subnetwork separately, we obtain a polynomial time algorithm. □

We briefly mention that our definition of Stackelberg network pricing is essentially equivalent to multi-product pricing with general valuation functions, a problem that has independently been considered in [BBM07]. Every general valuation function can be expressed in terms of Stackelberg network pricing on graphs of polynomial size, if we do not require the optimization problem of the follower to be polynomial time solvable. Our algorithmic results apply in this setting, as well, if we assume oracle access to the consumer valuations.

7.2 General Stackelberg Games and the Single-Price Algorithm - Again

We start by considering general Stackelberg pricing with a single follower. Surprisingly, the very simple single-price algorithm turns out to be successfully applicable once again. Remember that in case of unit-demand or single-minded consumers, it was straightforward to determine a set of candidate prices to be checked by the algorithm. This issue turns out to be a little more of a problem now.

Let c_0 denote the cost of a cheapest feasible subnetwork for the follower not containing any of the pricable edges. Clearly, we can compute c_0 by assigning price $+\infty$ to all pricable edges and simulating the follower on the resulting network. The single-price algorithm for Stackelberg network pricing proceeds as follows. For $j = 0, \dots, \lceil \log_{1+\varepsilon} c_0 \rceil$ it assigns price $p_j = (1+\varepsilon)^j$ to all pricable edges and determines the resulting revenue $r(p_j)$. It then simply returns the pricing that results in maximum revenue. We present a logarithmic bound on the approximation guarantee of the single-price algorithm.

Theorem 7.2.1. *For any $\varepsilon > 0$, the single-price algorithm for Stackelberg network pricing computes an $(1+\varepsilon)H_m$-approximation with respect to the optimal revenue $r(p^*)$.*

7.2.1 Analysis

The main issue in analyzing the algorithm's performance guarantee for Stackelberg pricing is to determine the right set of candidate prices. We first derive a precise characterization of these candidates and then argue that the geometric sequence of prices tested by the algorithm is a good enough approximation.

Slightly abusing notation, we let p refer to both price p and the assignment of this price to all pricable edges. If there exists a feasible subnetwork for the follower that uses at least j pricable edges, we let

$$\theta_j = \max\Big\{ p \,\Big|\, |S^*(p) \cap E_p| \geq j \Big\}$$

be the largest price at which such a subnetwork is chosen. If no feasible subnetwork with at least j pricable edges exists, we set $\theta_j = 0$. As we shall see, these thresholds are the key to prove Theorem 7.2.1.

We want to derive an alternative characterization of the threshold values θ_j. For each $1 \leq j \leq m$ we let c_j refer to the minimum sum of prices of fixed-price edges in any feasible subnetwork containing at most j pricable edges, formally

$$c_j = \min\Big\{ \sum_{e \in S \cap E_f} f_e \,\Big|\, S \in \mathcal{S} \,:\, |S \cap E_p| \leq j \Big\}.$$

Furthermore, for $1 \leq j \leq m$ we define

$$\Delta_j = c_0 - c_j,$$

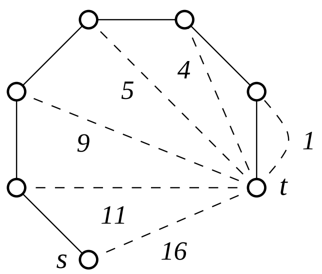
and, for ease of notation, let $\Delta_0 = 0$. Consider the point set $\{(0, \Delta_0), (1, \Delta_1), \dots, (m, \Delta_m)\}$ on the plane. By $\mathcal{H}$ we refer to a minimum selection of points spanning the upper convex hull of this point set. It is a straightforward geometric observation that we can define $\mathcal{H}$ as follows:

Fact 7.2.2. *Point (j, Δ_j) belongs to $\mathcal{H}$ if and only if*

$$\min_{i<j} \frac{\Delta_j - \Delta_i}{j - i} > \max_{j<k} \frac{\Delta_k - \Delta_j}{k - j}.$$

We now return to the candidate prices. By definition we have that $\theta_1 \geq \theta_2 \geq \dots \geq \theta_m$. We say that θ_j is *true threshold value* if $\theta_j > \theta_{j+1}$, i.e., if at price θ_j the subnetwork chosen by the follower contains exactly j pricable edges. Let $i_1 < i_2 < \dots < i_\ell$ denote the indices, such that θ_{i_k} are true threshold values and for ease of notation define $i_0 = 0$.

Lemma 7.2.3. *θ_j is true threshold value if and only if (j, Δ_j) belongs to $\mathcal{H}$.*



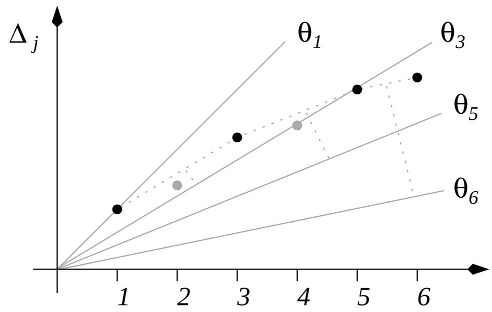


Figure 7.1: The follower wants to purchase a path connecting vertices s and t. Solid edges are pricable, dashed fixed price edges are labeled with their cost. The figure on the right gives a geometric interpretation of the corresponding (true) threshold values θ_j.

Proof: "⇒" Let θ_j be true threshold value, i.e., at price θ_j the chosen subnetwork contains exactly j pricable edges. We observe that at any price p the cheapest subnetwork containing i pricable edges has weight $c_i + ip = c_0 - \Delta_i + ip$. Thus, at price θ_j it must be the case that $\Delta_j - j\theta_j \geq \Delta_i - i\theta_j$ for all $i < j$ and $\Delta_j - j\theta_j > \Delta_k - k\theta_j$ for all $j < k$. It follows that

$$\min_{i<j} \frac{\Delta_j - \Delta_i}{j - i} \geq \theta_j > \max_{j<k} \frac{\Delta_k - \Delta_j}{k - j},$$

and, thus, we have that (j, Δ_j) belongs to $\mathcal{H}$.

"⇐" Assume now that (j, Δ_j) belongs to $\mathcal{H}$ and let

$$p \quad = \quad \min_{i<j} \frac{\Delta_j - \Delta_i}{j - i}. \tag{7.4}$$

Consider any $k < j$. It follows that

$$\Delta_k - kp = \Delta_j - jp - (\Delta_j - \Delta_k) + (j - k)p \leq \Delta_j - jp,$$

since $p \leq (\Delta_j - \Delta_k)/(j - k)$ and, thus, the network chosen at price p cannot contain less than j pricable edges. Analogously, let $k > j$. Using $p > (\Delta_k - \Delta_j)/(k - j)$ we obtain

$$\Delta_k - kp = \Delta_j - jp + (\Delta_k - \Delta_j) - (k - j)p < \Delta_j - jp,$$

and, thus, the subnetwork chosen at price p contains exactly j pricable edges. We conclude that θ_j is a true threshold. □

It is not difficult to see that the price p defined in equation (7.4) above is precisely the threshold value θ_j. We have already seen in the proof of Lemma 7.2.3 that at price p the follower chooses a subnetwork containing j pricable edges. Hence, we only need to argue that p is indeed maximal. Assume that $\ell < j$ minimizes (7.4), thus, $p = (\Delta_j - \Delta_\ell)/(j - \ell)$ and consider any $p' > p$. It holds that

$$\Delta_\ell - \ell p' = \Delta_j - jp' - (\Delta_j - \Delta_\ell) + (j - \ell)p' > \Delta_j - jp'$$

and, consequently,

$$c_\ell + \ell p' = c_0 - (\Delta_\ell - \ell p') < c_0 - (\Delta_j - jp') = c_j + jp'.$$

It follows that at price p' the cheapest feasible subnetwork contains less than j edges and the claim follows. Let now θ_{i_k} be a true threshold. Since points $(i_0, \Delta_{i_0}), \ldots, (i_\ell, \Delta_{i_\ell})$ define the convex hull we can write that

$$\min_{i<i_k} \frac{\Delta_{i_k} - \Delta_i}{i_k - i} = \frac{\Delta_{i_k} - \Delta_{i_{k-1}}}{i_k - i_{k-1}}.$$

We state this important fact again in the following lemma.

Lemma 7.2.4. *For all* $1 \leq k \leq \ell$ *it holds that*

$$\theta_{i_k} = \frac{\Delta_{i_k} - \Delta_{i_{k-1}}}{i_k - i_{k-1}}.$$

A geometric interpretation of true threshold values is found in Fig. 7.1. From the fact that points $(i_0, \Delta_{i_0}), \ldots, (i_\ell, \Delta_{i_\ell})$ define the upper convex hull we know that $\Delta_{i_\ell} = \Delta_m$, i.e., Δ_{i_ℓ} is the largest of all Δ-values. On the other hand, each Δ_j describes the maximum revenue that can be made from a subnetwork with at most j pricable edges and, thus, Δ_m is clearly an upper bound on the revenue made by an optimal price assignment.

Fact 7.2.5. *It holds that* $r(p^*) \leq \Delta_{i_\ell}$.

By definition of the θ_j's it is clear that at any price below θ_{i_k} the subnetwork chosen by the follower contains no less than i_k pricable edges. Furthermore, for each θ_{i_k} the single-price algorithm tests a candidate price that is at most a factor $(1+\varepsilon)$ smaller than θ_{i_k}.

Fact 7.2.6. *For each* θ_{i_k} *there exists a price* p_{i_k} *with* $(1+\varepsilon)^{-1}\theta_{i_k} \leq p_{i_k} \leq \theta_{i_k}$ *that is tested by the single-price algorithm. Especially, it holds that* $r(p_{i_k}) \geq (1+\varepsilon)^{-1} r(\theta_{i_k})$.

Finally, we know that the revenue made by assigning price θ_{i_k} to all pricable edges is $r(\theta_{i_k}) = i_k \theta_{i_k}$. Let p denote the price returned by the single-price algorithm, $r(p)$ its revenue. We have:

$$\begin{aligned}
(1+\varepsilon) \cdot H_m \cdot r(p) &= (1+\varepsilon) \sum_{j=1}^{m} \frac{r(p)}{j} = (1+\varepsilon) \sum_{k=1}^{\ell} \sum_{j=i_{k-1}+1}^{i_k} \frac{r(p)}{j} \\
&\geq (1+\varepsilon) \sum_{k=1}^{\ell} \sum_{j=i_{k-1}+1}^{i_k} \frac{r(p_{i_k})}{j} \geq \sum_{k=1}^{\ell} \sum_{j=i_{k-1}+1}^{i_k} \frac{r(\theta_{i_k})}{j} \\
&= \sum_{k=1}^{\ell} \sum_{j=i_{k-1}+1}^{i_k} \frac{i_k \theta_{i_k}}{j} \\
&\geq \sum_{k=1}^{\ell} (i_k - i_{k-1}) \frac{i_k \theta_{i_k}}{i_k} \\
&= \sum_{k=1}^{\ell} (\Delta_{i_k} - \Delta_{i_{k-1}}) \text{ , by Lemma 7.2.4} \\
&= \Delta_{i_\ell} - \Delta_0 = \Delta_{i_\ell} \geq r(p^*).
\end{aligned}$$

This concludes the proof of Theorem 7.2.1.

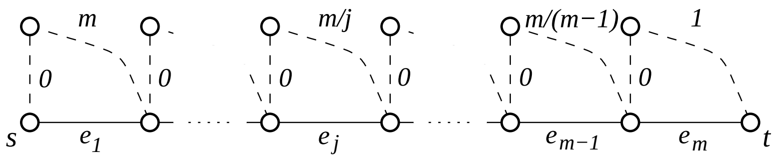


Figure 7.2: An instance of Stackelberg network pricing, on which the analysis of the approximation guarantee of the single-price algorithm is tight for the shortest path and minimum spanning tree case. Solid edges are pricable, dashed fixed-price edges are labeled with their cost.

7.2.2 Tightness

The example in Figure 7.2.1 shows that our analysis of the single-price algorithm's approximation guarantee is tight. The follower wants to buy a path connecting vertices s and t. In an optimal solution we set the price of edge e_j to m/j. Then edges $e_1, \ldots, e_m$ form a shortest path of cost mH_m. On the other hand, assume that all edges $e_1, \ldots, e_m$ are assigned the same price p. If $p \leq 1$ the leader's revenue is clearly bounded by m, if $p > m$ the shortest path does not contain any pricable edge at all. Let then $m/(j+1) < p \leq m/j$ for some $1 \leq j \leq m-1$. It is straightforward to argue that at this price a shortest path from s to t does not contain any of the pricable edges $e_{j+1}, \ldots, e_m$ and, thus, it contains at most j pricable edges. It follows that the leader's revenue is at most $j \cdot p \leq m$. Similar argumentation holds if the follower seeks to purchase a minimum spanning tree instead of a shortest path.

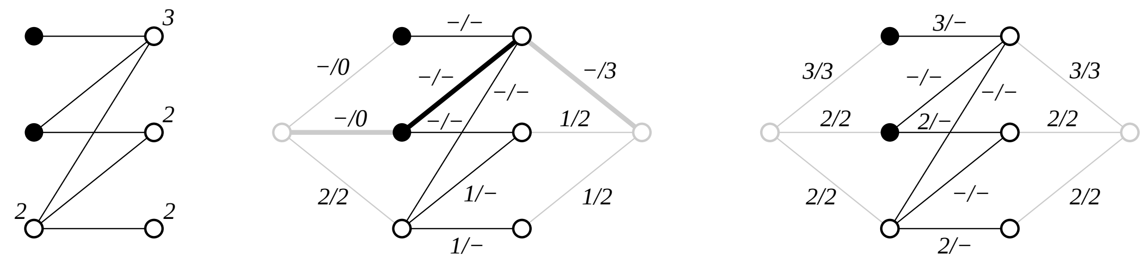
The best known lower bound for 2-player Stackelberg pricing is found in [CDF$^+$07], where APX-hardness is shown for the minimum spanning tree case. Still, for none of the Stackelberg pricing games considered so far could better than logarithmic approximation guarantees be proven. We proceed by introducing a special kind of Stackelberg pricing for which we can break this barrier.

7.3 Bipartite Stackelberg Vertex Cover

Bipartite Stackelberg Vertex Cover Pricing (StackVC) is a vertex game, in which we may assign prices to a subset $V_p \subset V$ of the vertices of a bipartite graph and players seek to purchase vertex covers for their respective subsets of edges. In general, the vertex cover problem is hard and, thus, we focus on settings in which the problem can be solved in polynomial time. In bipartite graphs the problem can be solved optimally by using a classical and fundamental max-flow/min-cut argumentation [PS98]. In principle, this does not imply anything for the complexity of the corresponding Stackelberg pricing game, as we will see in Section 7.4.2. However, when we restrict ourselves to the 2-player case, the pricing problem becomes tractable.

We first focus on the special case that all pricable vertices are on one side of the bipartition and prove that this problem is exactly solvable in polynomial time. We then present a natural extension of our algorithm that yields a 2-approximation in general.

Theorem 7.3.1. *If for a bipartite graph $G = (A \cup B, E)$ we have $V_p \subseteq A$, then there is a polynomial time algorithm computing an optimal price assignment p^* for 2-player* StackVC.



(a) (b) (c)

Figure 7.3: Solving 2-player STACKVC with pricable vertices in one partition and a single follower. Shaded vertices are pricable, vertex labels indicate cost, edge labels indicate flows and capacities. (a) A bipartite graph G. (b) The flow network G_d obtained from G with some initial flow. Grey parts are source and sink added by the transformation. Bold edges mark an augmenting path obtained by increasing the capacity on the leftmost edge. (c) The optimal flow at the end of the algorithm. Edges corresponding to pricable vertices are tight, the vertices belong to a minimum cost vertex cover.

We denote $n = |V_p|$ and again use the values c_j for $1 \leq j \leq n$ to denote the minimum sum of prices of fixed-price vertices in any vertex cover containing at most j pricable vertices. Then $\Delta_j = c_0 - c_j$ are again upper bounds on the revenue that can be extracted from a network that includes at most j pricable vertices and, thus, we have that $r(p^*) \leq \Delta_n$.

Algorithm 6: Solving 2-player STACKVC in bipartite graphs with $V_p \subseteq A$.

Construct the flow network G_d by adding nodes s and t.
Set $p(v) = 0$ for all $v \in V_p$.
Compute a maximum s-t-flow ϕ in G_d.
while *there is* $v \in V_p$ *s.t. increasing* $p(v)$ *yields an augmenting path* P **do**
    Increase $p(v)$ and ϕ along P as much as possible.

Suppose all pricable vertices are located in one partition, i.e., $V_p \subseteq A$, and consider Algorithm 6. Recall that for a bipartite graph G the LP-dual of the vertex cover problem can be captured by a maximum flow problem on an adjusted flow network G_d constructed as follows. We add a source s and a sink t to G and connect s to all vertices $v \in A$ with directed edges (s, v), and t to all vertices $v \in B$ with directed edges (v, t). Each such edge gets as capacity the price of the incident vertex, i.e., $p(v)$ for $v \in V_p$ or $c(v)$ if $v \in V_f$. Furthermore, we direct all original edges of the graph from A to B and set their capacity to infinity. It is well-known that the maximum s-t-flow in this network equals the cost of a minimum cost vertex cover of the graph G [PS98].

An augmenting path in G_d is a path traversing only forward edges with slack capacity and backward edges with non-zero flow. Algorithm 6 works by repeatedly increasing the prices of pricable vertices and the capacity of their incident edges in the flow network in order to create augmenting paths from s to t along which the flow can be increased. By LP-duality, the optimum vertex cover includes a vertex $v \in A$ if the maximum flow allows no augmenting path from s to v. For an illustration see Fig. 7.3. We denote by $\mathcal{V}$

the cover calculated by Algorithm 6.

Now consider a run of the algorithm. When computing the maximum flow on G_d holding all $p(v) = 0$, we get a flow of c_n, the cost of a cheapest vertex cover that may use all pricable vertices for free. We first note that in the following while-loop we will never face a situation, in which there is an augmenting s-t-path starting with a fixed-price vertex. We call such a path a *fixed* path, while an augmenting s-t-path starting with a pricable vertex is called a *price* path.

Lemma 7.3.2. *Every augmenting path considered in the while-loop of Algorithm 6 is a price path.*

Proof: We prove the lemma by induction on the while-loop and by contradiction. Suppose that in the beginning of the current iteration there is no fixed path. In particular, this is true for the first iteration of the while-loop. Then, suppose that after we have increased the flow over a price path P_p, a fixed path P_f is created. P_f must include some of the edges of P_p. Consider the vertex w at which P_f hits P_p. By following P_f from s to w and P_p from w to t there is a fixed path, which must have been present before flow was increased on P_p. This is a contradiction and proves the lemma. □

Note, that we may include a vertex $v \in A$ into the cover $\mathcal{V}$ if there is no augmenting path from s to v. In particular, this means that for a vertex $v \in A \cap \mathcal{V}$ the following two properties are fulfilled:

1. The flow over edge (s, v) equals the capacity and
2. there is no augmenting path from s over a different vertex $v' \in A$ that reaches v by decreasing flow over one of the original edges (v, w) for $w \in B$.

As the algorithm always adjusts the price of a vertex v to equal the current flow on (s, v), we can assume that there is never any slack capacity on edges (s, v) for any $v \in V_p$. Thus, only the violation of property 2 can force a vertex $v \in V_p$ to leave the cover. In particular, such an augmenting path must start with a fixed-price vertex. We call such a path a fixed v-path.

Lemma 7.3.3. *Algorithm 6 creates no fixed v-path for any pricable vertex $v \in V_p$.*

Proof: The proof is similar to the proof of the previous lemma. Suppose in the beginning of an iteration there is no fixed path, and additionally for a vertex $v \in V_p$ there is no fixed v-path. Then suppose such a path P_f^v is created by increasing flow over a price path P_p. Note that P_f^v cannot include any edge from P_p, because this would create a fixed path P_f as noted in Lemma 7.3.2. Furthermore, v must be included in P_p, because otherwise P_f^v would have existed initially. Now we can again use the same argument as before. Create a fixed path by following P_f^v from s to v and then P_p from v to t. This yields that a fixed path must have existed initially, which is a contradiction to the assumption. □

As there is no augmenting path from s to any pricable vertex at any time during the execution of the algorithm, the following lemma is now obvious.

Lemma 7.3.4. *$\mathcal{V}$ contains all pricable vertices.*

Finally, we are ready to prove Theorem 7.3.1 and argue that the computed price assignment is optimal.

Proof of Theorem 7.3.1: Suppose that after the execution of Algorithm 6 we increase price $p(v)$ beyond $\phi(s, v)$ for all pricable vertices $v \in V_p$. As we are at the end of the algorithm, it is not possible to increase the flow in the same way. Thus, the adjustment creates slack capacity on edge (s, v) for every $v \in V_p$ and causes all pricable vertices to leave $\mathcal{V}$. The new cover must be the cheapest cover that excludes all pricable vertices. Thus, it must have cost c_0. As we have not increased the flow, we know that the cost of $\mathcal{V}$ is also c_0. Note, that before starting the while-loop the cover contained all pricable vertices at price 0 and had cost c_n. As all flow increase in the while-loop was made over price paths and all the pricable vertices stay in the cover, the revenue of $\mathcal{V}$ must be $c_0 - c_n = \Delta_n$. This is an upper bound on the optimum revenue and, hence, the price assignment p computed by the algorithm is optimal. Finally, notice that adjusting the price of the pricable vertices in each iteration is not necessary. We can start with computing the initial cover containing all pricable vertices at price 0 and for the remaining while-loop set all prices to $+\infty$. This will result in the desired flow, which directly generates the final price for every vertex v as flow on edge (s, v). Hence, we can get optimal prices with an adjusted run of the standard polynomial time algorithm for maximum flow in G_d. This proves Theorem 7.3.1. □

Algorithm 7: A 2-approximation algorithm for 2-player STACKVC in bipartite graphs.

Fix $p_A(v) = \infty$ for all $v \in V_p \cap B$.
Fix $p_B(v) = \infty$ for all $v \in V_p \cap A$.
Run Algorithm 6 to determine $p_A(v)$ for $v \in V_p \cap A$.
Run Algorithm 6 to determine $p_B(v)$ for $v \in V_p \cap B$.
Return p_A or p_B, depending on which one yields higher revenue.

A natural extension of Algorithm 6 can be used to obtain a 2-approximation for general 2-player STACKVC. Algorithm 7 works by applying Algorithm 6 to the pricable vertices on each side of the bipartition separately in order to extract maximum revenue from one of these subsets.

Theorem 7.3.5. *Algorithm 7 is a 2-approximation algorithm for 2-player* STACKVC*. Furthermore, this bound is tight.*

Proof: Note that by setting $p_A(v) = \infty$ for all pricable vertices in B, we increase their price over the prices in the optimum solution. This obviously allows us to extract more revenue from the vertices in A than is possible for p^*. The same argument applies for the vertices in B and p_B. Hence, the sum of both revenues is an upper bound on $r(p^*)$, and our algorithm guarantees a 2-approximation by preserving the greater of the two.

For a tight example consider a path $(v_1, v_2, v_3, v_4, v_5)$. The first vertex v_1 is a pricable vertex, then there are two fixed-price vertices v_2 and v_3 of cost 1 and 0, respectively. Vertex v_4 is pricable again, v_5 has fixed cost 1. The optimum prices are $p(v_1) = p(v_3) = 1$. This yields the min-cost cover $\mathcal{V}^* = \{v_1, v_3, v_4\}$ and generates a revenue of 2. A possible solution returned by the algorithm, however, is $p(v_1) = 1$ and $p(v_2) = \infty$ (or vice versa), which generates only a revenue of 1. □

We next take a look at how the situation changes if we are faced with more than a single follower. Especially, in Section 7.4.1 we shall see that the complexity of STACKVC increases drastically in this situation.

7.4 Multi-Follower Stackelberg Pricing

Recall that each follower i is characterized by her own collection $\mathcal{S}_i$ of feasible subnetworks and k denotes the number of followers. Section 7.4.1 extends the analysis from the single follower case to prove a tight bound of $(1+\varepsilon)(H_k+H_m)$ on the approximation guarantee of the single-price algorithm and presents a matching lower bound based on STACKVC with multiple followers. Section 7.4.2 presents an alternative analysis that applies even in the case of players with different demands and yields approximation guarantees that do not depend on the number of followers. Furthermore, it contains a lower bound for the weighted case based on an instance of the river tarification problem.

7.4.1 General Unweighted Games

Let an instance of Stackelberg network pricing with some number $k \geq 1$ of followers be given. We extend the analysis from Section 7.2 to obtain a similar bound on the single-price algorithm's approximation guarantee. Notably, this immediately implies the previously known result for the single-minded case ([GHK$^+$05], cf. Theorem 6.2.1) up to a factor of $(1+\varepsilon)$.

Theorem 7.4.1. *The single-price algorithm computes an* $(1+\varepsilon)(H_k+H_m)$*-approximation for* STACK *with multiple followers with respect to the optimal revenue* $r(p^*)$*.*

Proof: Consider graph $G=(V,E)$, $E=E_f \cup E_p$ with $|E_p|=m$, and k followers defined by collections $\mathcal{S}_1,\ldots,\mathcal{S}_k$ of feasible subnetworks. We transform this instance into a single follower pricing game as follows. Let $G_1,\ldots,G_k$ be identical copies of G and define $G^* = G_1 \cup \ldots \cup G_k$. Furthermore, define a single follower by

$$\mathcal{S}^* = \{S_1 \cup \ldots \cup S_k \,|\, S_1 \in \mathcal{S}_1 \cap G_1, \ldots, S_k \in \mathcal{S}_k \cap G_k\},$$

i.e., for every follower i in the original instance our new follower seeks to purchase a subnetwork from $\mathcal{S}_i$ in copy G_i of the original graph. Clearly, the maximum possible revenue in the new instance is an upper bound on the maximum revenue in the multiple follower case, since we can always assign the same price to every copy of a pricable edge in $G_1,\ldots,G_k$. Furthermore, every pricing returned by the single-price algorithm on $G_1 \cup \ldots \cup G_k$ translates naturally into a corresponding pricing of identical revenue in G, since again all copies of an edge from G are assigned identical prices. Finally, since the number of pricable edges in $G_1 \cup \ldots \cup G_k$ is km, we obtain an approximation ratio of $(1+\varepsilon)H_{km}$ by Theorem 7.2.1 as desired. □

The reduction from the multiple to single follower case in the proof of Theorem 7.4.1 relies essentially on the fact that we are considering the single-price algorithm. More precisely, only the fact that all edges are assigned identical prices allows us to interpret price assignments on graph G^* as meaningful price assignments on the original graph G. Thus, the above does not imply anything about the relation between the single- and multiple-follower cases in general.

We next show an essentially tight lower bound, which follows immediately from the hardness of single-minded pricing (see Theorem 6.2.3). We encode SMP in terms of STACKVC with multiple followers.

Theorem 7.4.2. STACK *with multiple followers is hard to approximate within* $\mathcal{O}(\log^{\varepsilon} k + \log^{\varepsilon} m)$ *for some* $\varepsilon > 0$*, unless* NP $\subseteq \bigcap_{\delta>0}$ BPTIME($2^{\mathcal{O}(n^{\delta})}$)*. The same holds for* STACKVC *with multiple followers.*

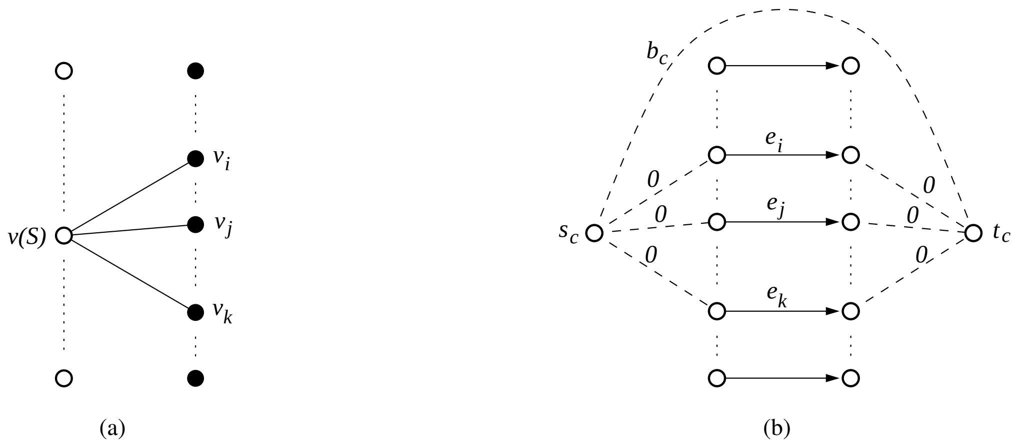


Figure 7.4: (a) Reducing SMP to STACKVC. The fixed-price vertex corresponding to set $S = \{u_i, u_j, u_k\}$ is assigned cost $v(S)$ and is adjacent to vertices representing products contained in S. (b) Reducing uniform-budget UDP($\mathcal{D}$)-MIN to the river tarification problem. Consumer c with $S_c = \{u_i, u_j, u_k\}$ is simulated by a follower routing her demand over one of the corresponding edges e_i, e_j, e_k. An alternative fixed-price route determines the (uniform) budget b_c.

Sketch of Proof: Let an instance of SMP with sets $S_1, \ldots, S_k$ over universe $\mathcal{U} = \{u_1, \ldots, u_m\}$ and corresponding valuations $v(S_i)$ be given. We define an instance of STACKVC as follows. In vertex set A we have a fixed-price vertex v_i for every set S_i with cost $c(v_i) = v(S_i)$. In vertex set B we have a pricable vertex v_j for every product $u_j \in \mathcal{U}$. Two vertices are connected by an edge if the corresponding set contains the respective product, formally, $(v_i, v_j) \in E$ if and only if $u_j \in S_i$. Finally, we have one follower for very set S_i who seeks to purchase a vertex cover of the edges adjacent to vertex $v_i \in A$. This construction is depicted in Fig. 7.4(a). □

7.4.2 General Weighted Games and the River Tarification Problem

We now turn to an even more general variation of Stackelberg pricing, in which we allow multiple *weighted* followers. This model, which has been previously considered in [BGvH+04], arises naturally in the context of network pricing games with different demands for each player. Formally, for each follower i we are given her *demand* $d_i \in \mathbb{R}^+$. For given prices p and optimal subnetworks $S_1^*(p), \ldots, S_k^*(p)$ of the followers, the leader's revenue is defined as

$$r(p) = \sum_{i=1}^{k} d_i p(S_i^*(p)).$$

It has been conjectured before that in the weighted case no approximation guarantee essentially beyond $\mathcal{O}(k \cdot \log m)$ is possible [RSM05]. We show that an alternative analysis of the single-price algorithm yields ratios that do not depend on the number of followers.

Theorem 7.4.3. *The single-price algorithm computes an $(1+\varepsilon)m^2$-approximation for* STACK *with multiple weighted followers with respect to the optimal revenue $r(p^*)$.*

Proof: Let graph $G = (V, E)$, $E = E_f \cup E_p$ with $|E_p| = m$, and k followers defined by $\mathcal{S}_1, \ldots, \mathcal{S}_k$ and demands $d_1, \ldots, d_k$ be given and consider the optimal pricing p^*. For each pricable edge, let $F(e)$ refer to the set of followers purchasing e under price assignment p^* and denote by $r(p^*|e) = \sum_{i \in F(e)} d_i p^*(e)$ the corresponding revenue. Clearly, $\sum_{e \in E_p} r(p^*|e) = r(p^*)$.

Fix some pricable edge e and define a corresponding price $p_e = p^*(e)/m$. By $r(p_e)$ we denote the revenue from assigning price p_e to all pricable edges. Let $i \in F(e)$ and assume that follower i buys subnetwork S_i under price assignment p^*. By $w^*(S_i)$, $w_e(S_i)$ and $c(S_i)$ we refer to the total weight of S_i under price assignments p^* and p_e and the cost due to fixed price edges only, respectively. It holds that

$$w_e(S_i) \leq c(S_i) + m\frac{p^*(e)}{m} = c(S_i) + p^*(e) \leq w^*(S_i).$$

Let c_i^0 denote the cost of a cheapest feasible subnetwork for follower i consisting only of fixed price edges. It follows that $w_e(S_i) \leq w^*(S_i) \leq c_i^0$ and, thus, follower i is going to purchase a subnetwork containing at least one pricable edge under price assignment p_e, resulting in revenue at least $d_i p_e = d_i p^*(e)/m$ from this follower. We conclude that $r(p_e) \geq r(p^*|e)/m$ and, thus

$$m^2 \max_{e \in E_p} r(p_e) \geq m \sum_{e \in E_p} r(p_e) \geq \sum_{e \in E_p} r(p^*|e) = r(p^*).$$

Finally, observe that for each price p_e the single-price algorithm checks some candidate price that is smaller by at most a factor of $(1 + \varepsilon)$, which finishes the proof. □

Hardness of approximation of Stackelberg pricing with weighted followers follows immediately by a reduction from distribution-based UDP($\mathcal{D}$)-MIN with uniform budgets (see Theorem 5.2.13). The resulting Stackelberg pricing game is an instance of the so-called *river tarification problem*, in which each player needs to route her demand along one out of a number of parallel links connecting her respective source and sink pair. One direct fixed price connection determines her maximum budget for purchasing a pricable link. Theorem 7.4.4 resolves an open problem from [BGvH$^+$04].

Theorem 7.4.4. STACK *with multiple weighted followers is hard to approximate within $\mathcal{O}(m^\varepsilon)$ for some $\varepsilon > 0$, unless* NP $\subseteq \bigcap_{\delta>0}$ BPTIME($2^{\mathcal{O}(n^\delta)}$). *The same holds for the river tarification problem.*

Sketch of Proof: Let an instance of uniform-budget UDP($\mathcal{D}$)-MIN with products $\mathcal{U} = \{u_1, \ldots, u_m\}$, consumer space $\mathcal{C} = \{c_1, \ldots, c_k\}$ and finite-support distribution $\mathcal{D}$ on $\mathcal{C}$ be given. For every product $u_j \in \mathcal{U}$ we have two vertices v_j^1, v_j^2 and a directed pricable edge (v_j^1, v_j^2). For every consumer c_i we have two vertices s_c, t_c and fixed-price edges (s_c, v_j^1), (v_j^2, t_c) of cost 0 for all u_j with $u_j \in S_c$. Additionally, there is a fixed-price edge (s_c, t_c) with fixed cost b_c. Now we introduce a follower i for every $c_i \in \mathcal{C}$ who wants to route a demand of $d_i = \Pr_{\mathcal{D}}(c_i)$ from s_c to t_c. The construction is illustrated in Fig. 7.4(b). □

7.5 Literature

The original definition of Stackelberg games to model economic processes dates back to the 1930's and Heinrich Freiherr von Stackelberg [vS34]. Computer scientists and especially the operations research community have been studying the problem of Stackelberg pricing in networks for about a decade.

Labbé et al. [LMS98] initiated the study of the shortest path version of the problem from an algorithmic perspective, which was then continued by Roch et al. [RSM05], who designed the first polynomial time approximation algorithms. An overview of most of the initial work in this area is found in [vH06].

Inspired by the minimum spanning tree version of Stackelberg pricing, Cardinal et al. [CDF$^+$07] apply the single-price algorithm to the problem and obtain provable approximation guarantees for all pricing games that are based on matroids, i.e., those pricing games in which the feasible subnetworks of a follower form the basis of a matroid. They also prove the first inapproximability result for any 2-player version of Stackelberg network pricing by deriving APX-hardness of the minimum spanning tree version. The multiple follower scenario appears first in [BGvH$^+$04], where also the river tarification problem is formulated.

Independently of our results from Section 7.2, Balcan et al. [BBM07] also analyze the performance of the single-price algorithm for pricing among consumers with general valuation functions and use this result to obtain improved guarantees for revenue maximization in some types of combinatorial auctions.

A different line of research has been investigating the application of Stackelberg strategies to network congestion games in order to obtain low congestion Nash equilibria for sets of selfish followers [CDR03, Rou04, Swa07]. Here, the leader is allowed to control prices on edges or the behavior of a fraction of the follower population in order to guarantee socially good Nash equilibria when the followers selfishly route their demands through the network.

The new results from this chapter have been published in [BHK07].

8 Conclusions and Future Research

We have presented results on the approximability of several different combinatorial multi-product and network pricing problems. For some of the established models the central questions have been answered to a satisfactory degree, for others the work has just begun.

For unit-demand pricing the current state of affairs leaves a relatively coherent picture of the problem's approximation complexity. Most notably, we could establish lower bounds on the approximation threshold of the well established min-buying model, which prove that the single-price algorithm essentially yields the best approximation guarantees we can hope for. Moreover, similar bounds appear to hold even in the uniform-budget case. Combinatorially speaking, this means that even in restricted cases of unit-demand pricing, all we can achieve is to select a single group of consumers that come with essentially identical budgets and set prices so as to extract full revenue from this group. The task of finding solutions that incorporate different groups of consumers and raise revenue across different price levels, however, is out of reach. In the words of Demaine et al. [DFHS06], this constitutes another example that "combination can be hard" - and indeed often is - in combinatorial pricing.

Even though quite a number of questions concerning unit-demand pricing have been resolved, still there are several intriguing open problems left for further research, out of which we will only name a few here. In Chapter 3 we have seen that the max-buying model with price-ladder constraint allows for a PTAS due to [AFMZ04], which also applies to rank-buying if we assume that budgets are rank-consistent. Thus, in the price-ladder scenario one can find arguably realistic selection rules that still allow practically relevant algorithmic results. It is an important open problem whether similar results can be obtained without a price-ladder constraint. More formally, what is the approximation threshold of rank-buying with rank-consistent budgets and without price-ladder? Note, that the important distinction here is "with rank-consistent budgets", as without this assumption we have shown that the problem is as hard as min-buying in Chapter 4. If this problem variation turns out to be hard to approximate, can one come up with new selection rules that do the trick, i.e., model rational consumer behavior while allowing for reasonably good (say constant) approximation guarantees? By our results from Chapter 4 we know that such selection rules cannot be based solely on the relative order of prices, but should be defined along the lines of rank-buying, in the sense that they incorporate some external description of consumers' preferences over products.

In Chapter 3 we present a local search algorithm that yields a 2-approximation for the max-buying model and works even in the more general case of limited product-supply. Aggarwal et al. [AFMZ04] show that their PTAS for UDP($\mathcal{C}$)-MAX-PL yields a 4-approximation for the limited-supply case with price-ladder constraint. Our local search approach is not applicable here, because a price-ladder constraint clearly restricts the way a solution can be modified locally. Still, in light of all other results on unit-demand pricing one would expect the price-ladder case to be no more difficult than the version without. Thus, it would be very interesting to see improved approximation guarantees for UDP($\mathcal{C}$)-MAX-PL with limited product supply. On the other hand, does the problem in fact become more difficult if supply is limited? Maybe it is possible to obtain stronger hardness results than the ones shown in Chapter 3 for unlimited supply.

Finally, the price-ladder constraint itself has turned out to be quite remarkable in the following sense. For general unit-demand pricing in the min-buying model we have seen that the presence of a price-ladder constraint does not have any influence on the problem's approximation threshold. Formally, both UDP($\mathcal{D}$)-MIN and UDP($\mathcal{D}$)-MIN-PL are hard to approximate within $\mathcal{O}(n^{1-\varepsilon})$ for all $\varepsilon > 0$, unless P = NP. In contrast, UDP($\mathcal{D}$)-MIN with uniform budgets has been proven to exhibit a comparable approximation threshold of $\mathcal{O}(n^{\varepsilon})$ for some $\varepsilon > 0$ in general, but becomes polynomial time solvable in the presence of a price-ladder constraint. Thus, a price-ladder can have quite different effects when applied to even slightly different versions of unit-demand pricing. This raises the following question, which is probably of rather theoretical interest: Are there versions of unit-demand pricing that are actually harder to approximate with price-ladder constraint than without? Maybe limited product supply is the key to answer this question. Apart from this, another natural and certainly relevant question, which is already being pursued in a lot of recent work, is this: Are there other restrictions than the price-ladder that allow for improved algorithmic results but capture practically relevant versions of the problem?

For the single-minded pricing problem the current picture is comparable to the unit-demand case. The question of approximability of the problem in general has essentially been settled. One open problem in this context is to obtain lower bounds on approximation guarantees in terms of the number of distinct products under standard complexity theoretic assumptions. We have seen in Chapter 6 that the problem is hard to approximate within $\mathcal{O}(n^{\varepsilon})$ for some $\varepsilon > 0$ under some assumption about the average-case complexity of refuting random 3SAT instances. We have also argued that similar results cannot follow from the reduction in [DFHS06], which yields pricing instances with only $\mathcal{O}(\log n)$ different price levels, which trivially allow for similar approximation guarantees.

Several open problems are left concerning the network pricing version of SMP. Foremost, we have presented an approximation algorithm that exploits the structure of sparse problem instances to achieve improved approximation guarantees. This algorithm, however, does not make any use of the (sparse) structure of the underlying network. Can we design algorithms that explicitly use the fact that they are operating on a network structure? Some results in this direction have been obtained for very specific networks as rooted trees [GHK$^+$05] or graphs consisting of a single line [ESZ07]. It would be very interesting to see similar results for more general network topologies.

The situation with Stackelberg pricing is quite different from the above. Some initial results have been proven both in this thesis and elsewhere, yet, the most central questions remain unanswered and many intriguing problems in the periphery of Stackelberg pricing games constitute quite promising ground for further research. As we have seen, the single-price algorithm can be used to achieve a logarithmic approximation guarantee for general 2-player Stackelberg pricing games but complementing lower bounds are rare. Roch et al. [RSM05] prove NP-hardness for the shortest path case, Cardinal et al. [CDF$^+$07] show that the Stackelberg minimum spanning tree game is APX-hard. The major open problem in the field is to prove a super-constant lower bound on the approximability of general 2-player Stackelberg pricing, as it appears quite unlikely that a constant approximation is possible in general, i.e., with a follower purchasing an arbitrary kind of subnetwork. Without having any strong opinion on the correctness of this observation, we mention that our findings so far indicate that the same should be true for the class of matroid Stackelberg games, in which the follower's feasible subnetworks form the basis of a matroid.

We have presented an alternative analysis of the single-price algorithm that proved an approximation guarantee of $\mathcal{O}(m^2)$ for the case of multiple weighted followers. We understand this result as somewhat conceptual, as it shows that contrary to previous conjectures it is possible to achieve approximation guarantees that do not depend on the number of followers in the game, but in our opinion is probably not a tight analysis. Thus, the question here is twofold. First, what is the actual approximation guarantee of the single-price algorithm for general Stackelberg pricing games with multiple weighted followers? Second, is it possible to achieve approximation ratio $\mathcal{O}(m)$? Note, that this would essentially match our lower bound for the river tarification problem.

The bipartite Stackelberg vertex cover game has turned out as an example of Stackelberg pricing in which the 2-player version allows for a constant approximation, and special cases can even be solved exactly in polynomial time. It is then a natural question to ask whether more such examples can be found. For example, one might consider games with a follower purchasing a min-cost matching, a min-cost cut, a min-cost cycle cover or several other natural types of subnetworks in the graph. We briefly mention that we have been able to obtain a similar result for a follower seeking to buy a min-cost knapsack cover (with bounded weights), which needs a little imagination to be viewed as a network pricing problem. However, interestingly the key to solve this problem is to use an extended version of the dynamic programming approach used to solve the min-knapsack problem itself. This is somewhat similar to the bipartite vertex cover case, where max-flow arguments were the key to success. More generally, an intriguing question is the following: Is there any clear connection between the combinatorial structure of the optimization problem solved by the follower and the complexity of the corresponding 2-player Stackelberg pricing game? Considering the minimum spanning tree case, it appears that there are cases in which the pricing problem is quite hard, even though the underlying optimization problem is of a quite well-structured type.

Finally, we have seen that at least for some combinatorial pricing problems local search yields reasonable approximation results. As we have argued before, local search appears to be a quite natural approach to multi-product pricing and it would be nice to have more examples of pricing problems to which local search, or more elaborate versions of it, can be successfully applied.

A Appendix

A.1 Complexity Classes, Reductions and Completeness

We summarize the definitions of the most important complexity classes. For a comprehensive introduction the reader is referred to [Pap94] or [Weg05]. The definitions presented here closely follow those in [Weg05].

Definition A.1.1. *A language $L \subseteq \{0,1\}^*$ belongs to the complexity class* P *(polynomial time), if there exists an algorithm A that outputs $A(x) = 1$ if and only if $x \in L$ and has running time poly(n) on inputs of length n.*

Definition A.1.2. *A language $L \subseteq \{0,1\}^*$ belongs to the complexity class* BPP *(bounded-error probabilistic polynomial time), if there exists a randomized algorithm A that has running time poly(n) on inputs of length n and satisfies the following for some constant $\varepsilon > 0$:*

1. $\Pr\big(A(x) = 1\big) \geq \frac{1}{2} + \varepsilon$ *if* $x \in L$
2. $\Pr\big(A(x) = 1\big) \leq \frac{1}{2} - \varepsilon$ *if* $x \notin L$

The complexity classes DTIME($t(n)$) and BPTIME($t(n)$) are defined just as P and BPP, respectively, with maximum running time poly(n) being replaced by $\mathcal{O}(t(n))$.

Definition A.1.3. *A language $L \subseteq \{0,1\}^*$ belongs to the complexity class* RP *(random polynomial time), if there exists a randomized algorithm A that has running time poly(n) on inputs of length n and satisfies the following for some constant $\varepsilon > 0$:*

1. $\Pr\big(A(x) = 1\big) \geq \varepsilon$ *if* $x \in L$
2. $\Pr\big(A(x) = 1\big) = 0$ *if* $x \notin L$

There are several equivalent ways to define complexity class NP. Definition A.1.4 follows [Weg05].

Definition A.1.4. *A language $L \subseteq \{0,1\}^*$ belongs to the complexity class* NP *(non-deterministic polynomial time), if there exists a randomized algorithm A that has running time poly(n) on inputs of length n and satisfies the following:*

1. $\Pr\big(A(x) = 1\big) > 0$ *if* $x \in L$
2. $\Pr\big(A(x) = 1\big) = 0$ *if* $x \notin L$

Definition A.1.5. *A language* $L_1 \subseteq \{0,1\}^*$ *is* polynomial time reducible *to a language* $L_2 \subseteq \{0,1\}^*$*, denoted* $L_1 \leq_p L_2$*, if there is a polynomial time computable function* $f : \{0,1\}^* \to \{0,1\}^*$*, such that for all* $x \in \{0,1\}^*$ *it holds that* $x \in L_1$ *if and only if* $f(x) \in L_2$*.*

Definition A.1.6. *A language* $L \in$ NP *is said to be* NP-complete*, if* $L' \leq_p L$ *for all* $L' \in$ NP*.*

We proceed by reviewing some standard complexity classes for optimization problems. The following definitions are taken from [Vaz03] and [Weg05]. An *optimization problem* Π consists of:

- D_Π: The set of valid instances. The *size* $|I|$ of an instance $I \in D_\Pi$ denotes the number of bits needed to encode the instance in binary.
- $S_\Pi(I)$: The set of feasible solutions for instance $I \in D_\Pi$.
- obj_Π: The objective function assigning values $\text{obj}_\Pi(I,s)$ to pairs of instances $I \in D_\Pi$ and solutions $s \in S_\Pi(I)$.

Finally, it has to be specified whether the objective function needs to be *minimized* or *maximized*. We let $\text{opt}_\Pi(I)$ denote the minimum or maximum objective function value obtainable on instance I, depending on whether Π is a minimization or maximization problem. Given optimization problem Π we can define the corresponding *decision problem* as the language $L_\Pi = \{(I,s) \mid I \in D_\Pi \wedge s \in S_\Pi(I)\}$. We say that Π is an *NP-optimization problem* if $L_\Pi \in$ P. For a minimization [maximization] problem Π the *approximation ratio* $r_\Pi(I,s)$ of a solution s to instance I is defined as:

$$r_\Pi(I,s) = \frac{\text{opt}_\Pi(I)}{\text{obj}_\Pi(I,s)} \qquad \left[r_\Pi(I,s) = \frac{\text{obj}_\Pi(I,s)}{\text{opt}_\Pi(I)} \right]$$

The approximation ratio (or *approximation guarantee*) of an algorithm A for optimization problem Π is defined as

$$r_A(n) = \sup\{r_\Pi(I, A(I)) \mid I : |I| \leq n\}.$$

In some places $r_A(n)$ and $r_A(n)^{-1}$ are used interchangeably for ease of notation. An algorithm with approximation ratio $r_A(n)$ is called an $r_A(n)$*-approximation algorithm* for Π. Problem Π is said to be *approximable within* $r_A(n)$ *(in polynomial time)*, if a (polynomial time) $r_A(n)$-approximation algorithm exists for Π.

Definition A.1.7. *An optimization problem* Π *belongs to the complexity class* APX*, if there exists an algorithm* A *with running time poly*$(|I|)$ *on input* I *and a constant* c*, such that* $A(I) \in S_\Pi(I)$ *and* $r_A(n) \leq c$ *for all* $I \in D_\Pi$*.*

Definition A.1.8. *A* PTAS reduction *of an optimization problem* Π_1 *to an optimization problem* Π_2*, denoted* $\Pi_1 \leq_{PTAS} \Pi_2$ *consist of a triple* (f, g, α) *of functions with the following properties:*

1. $f(I) \in D_{\Pi_2}$ *for all* $I \in D_{\Pi_1}$*. Furthermore,* f *is computable in polynomial time.*
2. $g(I, s, \varepsilon) \in S_{\Pi_1}(I)$ *for all* $s \in S_{\Pi_2}(f(I))$ *and* $\varepsilon \in \mathbb{Q}^+$*.*
3. $\alpha : \mathbb{Q}^+ \to \mathbb{Q}^+$ *is surjective and polynomial time computable.*

4. *If* $r_{\Pi_2}(f(I), s) \leq 1 + \alpha(\varepsilon)$*, then* $r_{\Pi_1}(I, g(I, s, \varepsilon)) \leq 1 + \varepsilon$.

Definition A.1.9. *An optimization problem* $\Pi \in$ APX *is said to be* APX-hard*, if* $\Pi' \leq_{PTAS} \Pi$ *for all* $\Pi' \in$ APX.

Definition A.1.10. *An algorithm A for optimization problem* Π *is called a* PTAS *(polynomial time approximation scheme), if on input* $I \in D_\Pi$ *and* $\varepsilon \in \mathbb{Q}^+$ *it has running time poly*$(|I|)$ *and produces a solution* $A(I) \in S_\Pi(I)$ *with* $r_\Pi(I, A(I)) \leq 1 + \varepsilon$.

Consequently, if any APX-hard problem Π had a PTAS, the same would be true for any problem in the class APX. By the PCP-theorem this would be equivalent to P = NP.

Definition A.1.11. *An algorithm A for optimization problem* Π *is called an* FPTAS *(fully polynomial time approximation scheme), if on input* $I \in D_\Pi$ *and* $\varepsilon \in \mathbb{Q}^+$ *it has running time poly*$(|I|, \varepsilon^{-1})$ *and produces a solution* $A(I) \in S_\Pi(I)$ *with* $r_\Pi(I, A(I)) \leq 1 + \varepsilon$.

A.2 Some Basics of Probability Theory

We briefly review some basics from probability theory and some important inequalities. For a thorough introduction to the field the reader is referred to [Fri96] or [MR95]. Recall that a *probability space* $(\Omega, \mathbb{F}, \Pr)$ consists of a *σ-field* $(\Omega, \mathbb{F})$ (i.e., a sample space Ω and a collection $\mathbb{F}$ of subsets of Ω that has $\emptyset$ as a member and is closed under complementation and countable unions) and a *probability measure* $\Pr : \mathbb{F} \to \mathbb{R}_0^+$. For an *event* $\mathcal{E} \in \mathbb{F}$, $\Pr(\mathcal{E})$ denotes its probability. For our purposes it is sufficient to think of $\mathbb{F}$ as 2^Ω. The following definitions and inequalities are taken from [MR95].

Definition A.2.1. *Let* $\mathcal{E}_1, \mathcal{E}_2 \in \mathbb{F}$*. The* conditional probability *of* $\mathcal{E}_1$ *given* $\mathcal{E}_2$ *is defined as*

$$\Pr(\mathcal{E}_1 \mid \mathcal{E}_2) = \frac{\Pr(\mathcal{E}_1 \cap \mathcal{E}_2)}{\Pr(\mathcal{E}_2)},$$

assuming that $\Pr(\mathcal{E}_2) > 0$.

Definition A.2.2. *A* random variable X *is a real valued function* $X : \Omega \to \mathbb{R}$*, such that for all* $x \in \mathbb{R}$ *we have that* $\{\omega \in \Omega \mid X(\omega) \leq x\} \in \mathbb{F}$.

Similar to the probability measure on $\mathbb{F}$ a density function describes the distribution of a random variable. We will restrict ourselves to *discrete random variables* whose range is either a finite or countably infinite subset of $\mathbb{R}$.

Definition A.2.3. *The* density function $p_X : \mathbb{R} \to [0, 1]$ *for a random variable* X *is defined as* $p_X(x) = \Pr(X = x)$*. Similarly, the* joint density function $p_{X,Y} : \mathbb{R} \times \mathbb{R} \to [0, 1]$ *of two random variables* X*,* Y *is defined as* $p_{X,Y}(x, y) = \Pr(X = x \wedge Y = y)$.

Definition A.2.4. *Random variables* X *and* Y *are said to be* independent *if for all* $x, y \in \mathbb{R}$,

$$p_{X,Y}(x, y) = \Pr(X = x) \cdot \Pr(Y = y).$$

Definition A.2.5. *The* expectation *of a random variable X with density function p_X is defined as*

$$\mathrm{E}[X] = \sum_x x p_X(x).$$

Lemma A.2.6. *Let $X_1, \ldots, X_n$ be random variables, $c_1, \ldots, c_n \in \mathbb{R}$. It holds that*

$$\mathrm{E}[\sum_{i=1}^{n} c_i X_i] = \sum_{i=1}^{n} c_i E[X_i].$$

This property is referred to as linearity of expectation*.*

Note that linearity of expectation does not require independence of the random variables. Finally, we state two important inequalities.

Theorem A.2.7 (Markov Inequality)**.** *Let X be a random variable assuming only non-negative values. Then, for all $t \in \mathbb{R}^+$, it holds that*

$$\Pr(X \geq t) \leq \frac{\mathrm{E}[X]}{t}.$$

Theorem A.2.8 states simplified versions of the Chernoff tail inequalities as used in Chapter 5.

Theorem A.2.8 (Chernoff Bound)**.** *Let $X_1, \ldots, X_n \in \{0, 1\}$ be independent random variables with $\Pr(X_i = 1) = p_i$ for $1 \leq i \leq n$. Then, for $X = \sum_{i=1}^{n} X_i$, $\mu = \mathrm{E}[X]$ and any $0 < \delta \leq 2\mathrm{e} - 1$, it holds that*

$$\Pr(X > (1 + \delta)\mu) \leq \mathrm{e}^{-\frac{\mu\delta^2}{4}}.$$

For any $\delta \geq 2\mathrm{e} - 1$ it holds that

$$\Pr(X > (1 + \delta)\mu) \leq \mathrm{e}^{-(1+\delta)\mu}.$$

Similarly, for any $0 < \delta \leq 1$ it holds that

$$\Pr(X < (1 - \delta)\mu) \leq \mathrm{e}^{-\frac{\mu\delta^2}{2}}$$

B List of Symbols

$\mathcal{C}$	a set of unit-demand consumer samples
$\mathcal{D}$	a distribution over unit-demand consumers
e	the Euler constant
$\mathrm{E}[X]$	expectation of the random variable X
$\mathrm{E}_D[X]$	expectation of the random variable X according to distribution D
G-SMP	the single-minded pricing problem on networks
H_k	the k'th harmonic number $\sum_{i=1}^{k} \frac{1}{i} \approx \ln k$
MES	the maximum expanding sequence problem
$\mathbb{N}$	the natural numbers
$\mathcal{O}(\cdot)$, $\Omega(\cdot)$, $o(\cdot)$, $\omega(\cdot)$	standard assymptotic notation
p	a price assignment
poly(n)	$\mathcal{O}(n^c)$ for some constant c
$\Pr(\mathcal{E})$	probability of event $\mathcal{E}$
$\Pr_D(\mathcal{E})$	probability of event $\mathcal{E}$ according to distribution D
$\mathbb{Q}$	the rationals
$\mathbb{R}$	the reals
$\mathbb{R}^+$	the positive reals
$\mathbb{R}_0^+$	the positive reals including 0
$r_s(p)$	the revenue of price assignment p in unit-demand pricing with selection rule s
R3SAT$^\star(t(n))$	see Definition 5.2.2
SMP	the single-minded pricing problem
$\mathcal{U}$	a universe of products
UDP($\mathcal{C}$)-s	unit-demand pricing with consumer samples $\mathcal{C}$ and selection rule s
UDP($\mathcal{D}$)-s	unit-demand pricing with consumer distribution $\mathcal{D}$ and selection rule s
UDP($\mathcal{C}$)-s-PL	unit-demand pricing with consumer samples $\mathcal{C}$, selection rule s and price-ladder
UDP($\mathcal{D}$)-s-PL	unit-demand pricing with consumer distribution $\mathcal{D}$, selection rule s and price-ladder
STACK	the general Stackelberg pricing game
STACKVC	the bipartite Stackelberg vertex cover pricing game

Bibliography

[ACG$^+$99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer, 1999.

[AFMZ04] G. Aggarwal, T. Feder, R. Motwani, and A. Zhu. Algorithms for Multi-Product Pricing. In *Proc. of 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 72–83, 2004.

[AFWZ95] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman. Derandomized Graph Products. *Computational Complexity*, 5(1):60–75, 1995.

[ALM$^+$98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and Hardness of Approximation Problems. *Journal of the ACM*, 45(3):501–555, 1998.

[AT01] A. Archer and E. Tardos. Truthful Mechanisms for One-Parameter Agents. In *Proc. of 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[BB06] N. Balcan and A. Blum. Approximation Algorithms and Online Mechanisms for Item Pricing. In *Proc. of 7th ACM Conference on Electronic Commerce (EC)*, pages 29–35, 2006.

[BBCH07] M. Balcan, A. Blum, H. Chan, and M. Hajiaghayi. A Theory of Loss-Leaders: Making Money by Pricing Below Cost. Technical Report CMU-CS-07-143, Carnegie Mellon University, 2007.

[BBHM05] N. Balcan, A. Blum, J. Hartline, and Y. Mansour. Mechanism Design via Machine Learning. In *Proc. of 46th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 605–614, 2005.

[BBM07] M. Balcan, A. Blum, and Y. Mansour. Single Price Mechanisms for Revenue Maximization in Unlimited Supply Combinatorial Auctions. Technical Report CMU-CS-07-111, Carnegie Mellon University, 2007.

[BCES06] N. Bansal, A. Chakrabarti, A. Epstein, and B. Schieber. A Quasi-PTAS for Unsplittable Flow on Line Graphs. In *Proc. of the 38th ACM Symposium on Theory of Computing (STOC)*, pages 721–729, 2006.

[BGvH$^+$04] M. Bouhtou, A. Grigoriev, S. van Hoesel, A. van der Kraaij, and M. Uetz. Pricing Network Edges to Cross a River. In *Proc. of 2nd Workshop on Approximation and Online Algorithms (WAOA)*, pages 140–153, 2004.

[BHK07] P. Briest, M. Hoefer, and P. Krysta. Stackelberg Network Pricing Games. Technical Report ULCS-07-022, The University of Liverpool, 2007.

[BK06] P. Briest and P. Krysta. Single-Minded Unlimited-Supply Pricing on Sparse Instances. In *Proc. of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1093–1102, 2006.

[BK07] P. Briest and P. Krysta. Buying Cheap is Expensive: Hardness of Non-Parametric Multi-Product Pricing. In *Proc. of 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 716–725, 2007.

[BKV05] P. Briest, P. Krysta, and B. Vöcking. Approximation Techniques for Utilitarian Mechanism Design. In *Proc. of 37th ACM Symposium on Theory of Computing (STOC)*, pages 39–48, 2005.

[Blu91] A. Blum. *Algorithms for Approximate Graph Coloring*. PhD thesis, MIT Laboratory for Computer Science, 1991. MIT/LCS/TR-506.

[Bri06] P. Briest. Towards Hardness of Envy-Free Pricing. Technical Report TR06-150, ECCC, 2006.

[BS92] P. Berman and G. Schnitger. On the Complexity of Approximating the Independent Set Problem. *Information and Computation*, 96(1):77–94, 1992.

[CCPS98] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, 1998.

[CDF$^+$07] J. Cardinal, E. Demaine, S. Fiorini, G. Joret, S. Langerman, I. Newman, and O. Weimann. The Stackelberg Minimum Spanning Tree Game. In *Proc. of 10th Workshop on Algorithms and Data Structures (WADS)*, pages 64–76, 2007.

[CDR03] R. Cole, Y. Dodis, and T. Roughgarden. Pricing Network Edges for Heterogeneous Selfish Users. In *Proc. of 35th ACM Symposium on Theory of Computing (STOC)*, pages 521–530, 2003.

[CHK07] S. Chawla, J. Hartline, and R. Kleinberg. Algorithmic Pricing via Virtual Valuations. In *Proc. of 8th ACM Conference on Electronic Commerce (EC)*, pages 243–251, 2007.

[CSS06] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[DFHS06] E. Demaine, U. Feige, M. Hajiaghayi, and M. Salavatipour. Combination Can Be Hard: Approximability of the Unique Coverage Problem. In *Proc. of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 162–171, 2006.

[EGL$^+$98] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velickovic. Efficient Approximation of Product Distributions. *Random Structures and Algorithms*, 13(1):1–16, 1998.

[ESZ07] K. Elbassioni, R. Sitters, and Y. Zhang. A Quasi-PTAS for Envy-Free Pricing on Line Graphs. In *Proc. of 15th Annual European Symposium on Algorithms (ESA)*, 2007. To appear.

[Fei02] U. Feige. Relations between Average Case Complexity and Approximation Complexity. In *Proc. of 34th ACM Symposium on Theory of Computing (STOC)*, pages 534–543, 2002.

[FHKS02] U. Feige, M. Halldorsson, G. Kortsarz, and A. Srinivasan. Approximating the Domatic Number. *SIAM Journal on Computing*, 32(1):172–195, 2002.

[FK04] U. Feige and S. Kogan. Hardness of Approximation of the Balanced Complete Bipartite Subgraph Problem. Technical Report MCS04-04, Dept. of Computer Science and Applied Mathematics, The Weizmann Institute of Science, 2004.

[FPT04] A. Fabrikant, C. Papadimitriou, and K. Talwar. The Complexity of Pure Nash Equilibria. In *Proc. of 36th ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.

[Fri96] B. Fristedt. *A Modern Approach to Probability Theory*. Birkhäuser, 1996.

[GHK+05] V. Guruswami, J. Hartline, A. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On Profit-Maximizing Envy-Free Pricing. In *Proc. of 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1164–1173, 2005.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[GRR06] P. Glynn, P. Rusmevichientong, and B. Van Roy. A Non-Parametric Approach to Multi-Product Pricing. *Operations Research*, 54(1):82–98, 2006.

[GvLS+07] A. Grigoriev, J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld. Bundle Pricing with Comparable Items. In *Proc. of the 15th Annual European Symposium on Algorithms (ESA)*, 2007. To appear.

[HK05] J. Hartline and V. Koltun. Near-Optimal Pricing in Near-Linear Time. In *Proc. of 9th Workshop on Algorithms and Data Structures (WADS)*, pages 422–431, 2005.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander Graphs and Their Applications. *Bulletin of the AMS*, 43:439–561, 2006.

[KB57] T. Koopmans and M. Beckmann. Assignment Problems and the Location of Economic Activities. *Econometrica*, 25:53–76, 1957.

[Kho02] S. Khot. On the Power of Unique 2-Prover 1-Round Games. In *Proc. of 34th ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002.

[Kho04] S. Khot. Ruling out PTAS for Graph Min-Bisection, Densest Subgraph and Bipartite Clique. In *Proc. of 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 136–145, 2004.

[KMR07] R. Krauthgamer, A. Mehta, and A. Rudra. Pricing Commodities, or How to Sell When Buyers Have Restricted Valuations. In *Proc. of 5th Workshop on Approximation and Online Algorithms (WAOA)*, 2007. To appear.

[KP99] E. Koutsoupias and C. Papadimitriou. Worst-Case Equilibria. In *Proc. of 16th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 404–413, 1999.

[LMS98] M. Labbé, P. Marcotte, and G. Savard. A Bilevel Model of Taxation and its Application to Optimal Highway Pricing. *Management Science*, 44(12):1608–1622, 1998.

[MN02] A. Mu'alem and N. Nisan. Truthful Approximation Mechanisms for Restricted Combinatorial Auctions, 2002. In AAAI (poster), 2002. Also presented at Dagstuhl Workshop on Electronic Market Design.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Mye81] R. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6:58–73, 1981.

[NR99] N. Nisan and A. Ronen. Algorithmic Mechanism Design. In *Proc. of 31st ACM Symposium on Theory of Computing (STOC)*, pages 129–140, 1999.

[Owe95] G. Owen. *Game Theory, 3rd Ed.* Academic Press, 1995.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pap01] C. Papadimitriou. Algorithms, Games, and the Internet. In *Proc. of 33rd ACM Symposium on Theory of Computing (STOC)*, pages 749–753, 2001.

[Pro] `http://www.myproductadvisor.com/`.

[PS98] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.

[PY91] C. Papadimitriou and M. Yannakakis. Optimization, Approximation and Complexity Classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Rou04] T. Roughgarden. Stackelberg Scheduling Strategies. *SIAM Journal on Computing*, 33(2):332–350, 2004.

[RSM05] S. Roch, G. Savard, and P. Marcotte. An Approximation Algorithm for Stackelberg Network Pricing. *Networks*, 46(1):57–67, 2005.

[Rus03] P. Rusmevichientong. *A Non-Parametric Approach to Multi-Product Pricing: Theory and Application*. PhD thesis, Stanford University, 2003.

[ST06] A. Samorodnitsky and L. Trevisan. Gowers Uniformity, Influence of Variables, and PCPs. In *Proc. of 38th ACM Symposium on Theory of Computing (STOC)*, pages 11–20, 2006.

[Swa07] C. Swamy. The Effectiveness of Stackelberg Strategies and Tolls for Network Congestion Games. In *Proc. of 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1133–1142, 2007.

[Vaz03] V. V. Vazirani. *Approximation Algorithms*. Springer, 2003.

[vH06] S. van Hoesel. An Overview of Stackelberg Pricing in Networks. Research Memoranda 042, METEOR, Maastricht, 2006.

[vS34] H. von Stackelberg. *Marktform und Gleichgewicht*. Verlag von Julius Springer, Wien, 1934.

[Weg05] I. Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Springer, 2005.

[Zuc06] D. Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. In *Proc. of 38th ACM Symposium on Theory of Computing (STOC)*, pages 681–690, 2006.