

Algorithmic pricing

Citation for published version (APA):

van Loon, J. (2009). *Algorithmic pricing*. [Doctoral Thesis, Maastricht University]. Universitaire Pers Maastricht. <https://doi.org/10.26481/dis.20090313jl>

Document status and date:

Published: 01/01/2009

DOI:

[10.26481/dis.20090313jl](https://doi.org/10.26481/dis.20090313jl)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Algorithmic Pricing

Joyce Maria Wilhelmina van Loon

This book was typeset by the author using LaTeX.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission in writing from the author.

Algorithmic Pricing

© Joyce Maria Wilhelmina van Loon, 2009

Cover design by Rainman.nl

Published by Universitaire Pers Maastricht

ISBN 978-90-5278-803-6

Printed in the Netherlands by Datawyse

Algorithmic Pricing

Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit Maastricht,
op gezag van Rector Magnificus,
Prof. mr. G.P.M.F. Mols,
volgens het besluit van het College van Decanen,
in het openbaar te verdedigen
op vrijdag 13 maart 2009 om 14:00 uur

door

Joyce Maria Wilhelmina van Loon



Promotores:

Prof. dr. M.J. Uetz (Universiteit Twente)

Prof. dr. ir. C.P.M. van Hoesel

Copromotor:

Dr. A. Grigoriev

Beoordelingscommissie:

Prof. dr. R.J. Müller (voorzitter)

Dr. A. Berger

Prof. dr. M. Labbé (Université Libre de Bruxelles)

Dit onderzoek werd financieel mogelijk gemaakt door Maastricht Research School of Economics of Technology and Organizations (METEOR).

Acknowledgements

This thesis would never have been completed without the help and encouragement of many people. I am truly thankful to my family, friends and colleagues who supported me during my time as a PhD student. At this point I would like to thank several persons who contributed in a special way.

First of all, I wish to thank my supervisors. I am grateful to Marc Uetz for his encouragement and advice, and for sharing his professional expertise and excellent writing skills. I owe special thanks to Alexander Grigoriev, whose enthusiasm and creativity are a source of inspiration. I am truly thankful for the stimulating environment that Marc and Alex created for me and for the time they spent to guide and help me.

I thank Stan van Hoesel for introducing me to several projects from which I benefited a lot. I am grateful to Tjark Vredeveld for his help with any problem that occurred during my research. Special thanks to André Berger for extensively reading a preliminary and the final version of the manuscript. His remarks helped greatly while writing the final version of this thesis. I thank Rudolf Müller and Martine Labbé for their time and willingness to read the manuscript and providing me with useful comments.

Special thanks to Haydeé Hallmanns and Karin van den Boorn for their help with all administrative problems, and for the pleasant and relaxing chats. I wish to thank my roommate Bert Marchal with whom I could share the good parts of being a PhD student but also all frustrations. I will certainly miss our coffee-time together.

I thank my paronyms, Birgit Heydenreich and Karin Lamberts. I wish to thank Birgit for her help and support as a fellow PhD student, but also for becoming a dear friend. I am thankful to Karin for many years of close

Acknowledgments

friendship, fun evenings and trips, and her moral support during the most difficult times.

I am very grateful to Jo, Clementine and Daniëlla Moonen for making me feel at home in their family.

My parents, Ruud and Anny, have always been a great source of support. Their unconditional love and pride gave me the confidence to achieve anything I wanted. I am greatly indebted to them for their help and encouragement.

My biggest thanks go to Raymond. I wish to truly thank him for his love, patience and for always being there for me. I feel very grateful to have him by my side.

Joyce van Loon
Maastricht, January 2009

Contents

1	Introduction	11
1.1	Algorithmic pricing	11
1.2	Single item pricing	13
1.2.1	Tollbooth and highway problem	15
1.3	Affine pricing	16
1.3.1	Price regulation	18
1.4	Related work	19
1.5	Outline	21
1.6	List of publications	24
I	Single Item Pricing	25
2	Monotonicity Constraint on Prices	27
2.1	Introduction	27
2.2	Arbitrary bundles	29
2.2.1	Complexity	29
2.2.2	Approximation scheme	33
2.3	Highway problem	35
2.3.1	Complexity	36
2.3.2	Approximation algorithm	43
2.4	Conclusion	45
3	Inhomogeneity of Average Valuations	47
3.1	Introduction	48

3.2	Arbitrary bundles	49
3.2.1	Complexity	49
3.2.2	Approximation algorithm	54
3.3	Highway problem	58
3.3.1	Complexity	59
3.4	Conclusion	63
4	Pricing in Capacitated Graphs	65
4.1	Introduction	65
4.2	Highway problem	67
4.2.1	Complexity	68
4.2.2	Approximation algorithm	69
4.2.3	Dynamic program	71
4.2.4	FPTAS	74
4.3	Pricing in a capacitated cycle	76
4.4	Unit supply	78
4.4.1	Highway problem	78
4.4.2	Cycle	79
4.4.3	Tree	80
4.4.4	Grid	81
4.5	Conclusion	83
II	Affine Pricing	85
5	Algorithms for Affine Pricing	87
5.1	Introduction	87
5.2	Affine pricing algorithm	89
5.2.1	Limited supply	92
5.2.2	Unit-demand customers	95
5.3	Local search	96
5.4	Computational study	101
5.4.1	Results	102
5.5	Conclusion	103
6	Price Strategy Implementation	105
6.1	Introduction	106
6.2	Implementation in minimal time	107
6.3	Maximize total revenue	110

6.3.1	Dynamic program	110
6.3.2	Local search	111
6.4	Computational results	114
6.5	Conclusion	116
7	Algorithms for Optimal Price Regulations	119
7.1	Introduction	120
7.1.1	Three-level program	121
7.2	Parameter and tax level optimization	124
7.3	Optimization of European regulation	129
7.3.1	Price regulation by tax	131
7.3.2	Parameter optimization	136
7.4	Computational results	137
7.4.1	Experiments	138
7.5	Conclusion	140
	Bibliography	143
	Index	151
	Nederlandse Samenvatting	153
	Curriculum Vitae	157

Chapter 1

Introduction

A central issue for every company is to price the items for sale with the objective to maximize the revenue. If pricing decisions are made without sufficient research and analysis, the company can lose a significant amount of revenue. A low price can lead to a loss if interested customers are willing to spend more to purchase the item. It will, however, also attract more customers that would otherwise not buy the item. A high price generates more revenue per sold item, but interested customers might decide not to purchase the item simply because it is too expensive. Setting the price level to maximize the revenue takes considerable market knowledge and algorithmic analysis.

1.1 Algorithmic pricing

The pricing problem is a two-level game between the company and the customers. First, the company sets the prices for the items with the objective to maximize the revenue. Then, regarding these prices, customers optimize their own objective, which means that they decide which items they want to purchase and whether they are willing to pay the price set by the company. This two-level game, where the company is the leader and the customers are the followers, is an instance of a Stackelberg game (von Stackelberg 1934).

Being the leader of the Stackelberg game, the company has to act first. Obviously, it is a great advantage to the company if it already has access to information about the preferences of the customers at the moment the prices are to be determined. Obtaining a thorough knowledge about the target market is nowadays easier and more distributed than in the past,

mainly through the use of the Internet (Rendleman 2001; Rusmevichientong, van Roy, and Glynn 2006). A company can buy data about almost every industry, product, trend and interest, in any manner that fits best to the company's specific needs. Because of this large availability and easiness to spread information, companies know what customers buy, how much they spend and how frequently a product is purchased. That information can be worth a lot of money if it is used to the fullest. In *algorithmic pricing*, we address the computational problem that a company encounters when determining the prices in order to maximize the revenue, using all information about the preferences of the customers.

Algorithmic pricing problems are also motivated by their importance in the field of algorithmic mechanism design¹. In mechanism design problems, the customers' preferences are private information and thus unknown to the company. The goal of the company is to create an incentive compatible mechanism, which means that for all customers the best strategy is to truthfully reveal their private information. This particular problem is discussed by Goldberg and Hartline (2001) and Goldberg et al. (2006). More specifically, they study competitive auctions, which are auctions that achieve a revenue that is a constant fraction of the optimal solution on every input of the problem. This optimal revenue can be found using algorithmic pricing. Particularly, a general approach for reducing incentive compatible mechanism design problems to the underlying algorithmic pricing problems is suggested by Balcan et al. (2005) and further explored in Balcan et al. (2008). Hence, there is interest in purely algorithmic pricing problems from the perspective of mechanism design, as in order to solve the algorithmic mechanism design problem, it is useful to understand how items should be priced when the preferences of the customers are known.

In this thesis we address two different algorithmic pricing problems; the single item pricing problem and the affine pricing problem. The first part of this thesis is dedicated to the *single item pricing* problem. In Chapters 2, 3 and 4, we address the complexity of this problem subject to several natural constraints, and develop algorithms to solve or approximate these problems. The second part is devoted to the *affine pricing* problem. We consider the computational complexity of this problem in Chapter 5, and describe practi-

¹For further reading on algorithmic mechanism design, we refer to Nisan and Ronen (1999), Mu'alem and Nisan (2002), Archer et al. (2003), Abrams (2006), Aggarwal and Hartline (2006), Balcan, Blum, and Mansour (2007), and a recent survey by Hartline and Karlin (2007).

cal settings in which this problem can be applied in Chapters 6 and 7. Next to the theoretical results, we conduct several computational studies in the last three chapters of this thesis.

The single item pricing problem and the affine pricing problem are introduced in the following two sections.

1.2 Single item pricing

In single item pricing, the company owns a set of items for sale and wants to determine a price for each item. Every customer requests one or more subsets of items, referred to as *bundles*. Each bundle contains at most one copy of every item. For every bundle a customer requests, she has a *valuation*, which is the maximum amount she is willing to pay for that particular bundle. The total price of a bundle is equal to the sum of the prices for all items contained in the bundle. A customer is only interested in a bundle if its price does not exceed her valuation. The company's goal is to maximize the revenue, which is equal to the total sum of prices of all items sold.

Customer profiles, and thus the way the requests are established, can be very different. The two most important customer profiles in algorithmic pricing problems are unit-demand and single-minded. A *unit-demand* customer requests many different bundles and has a separate valuation for each of those. Based on the prices set by the company, the customer determines the bundle that she prefers most, namely the one that yields the highest positive *utility* which is equal to her valuation minus the price of that particular bundle. In contrast to a unit-demand customer, a *single-minded* customer is interested in one particular bundle only. By definition, a customer is single-minded if and only if she has a positive valuation for one bundle, and values all other bundles equal to zero. She purchases the bundle with positive valuation if the total price does not exceed her valuation. Single-minded customers with private valuations are widely discussed in combinatorial auction theory². However, in algorithmic pricing problems, single-minded customers are introduced only recently.

Items can be available to customers in unlimited or limited supply. *Unlimited supply* means that there are either at least as many copies per item as there are customers, or that the items can instantly be produced without significant costs, which is for example generally the case for digital items.

²See e.g. Lehmann, O'Callaghan, and Shoham (2002), Mu'alem and Nisan (2002), Archer et al. (2003), and Cramton, Shoham, and Steinberg (2006).

Consequently, every customer that can afford to purchase a particular bundle is able to do so. In contrast, if items are available in *limited supply*, the allocation of bundles to customers is a much harder task. It might occur that some customers want to purchase a bundle but are unable to do so by the limitation of available items. The company determines the prices and allocates bundles of items to customers, and can choose to do this *envy-free*. In game theory, envy-freeness is a property of a *fair* allocation of items to customers who may have different preferences (Walras 1954; Gul and Stacchetti 1999). An allocation is said to be envy-free if no customer prefers another bundle to her own (Foley 1967). Envy-freeness is recently introduced in single item pricing (Aggarwal et al. 2004; Guruswami et al. 2005) to obtain a better algorithmic understanding of game theory. In an optimal envy-free pricing, the company has no incentive to change the prices on the items and the customers have no incentive to complain about the allocation, given the pricing. In the unit-demand case, this means that every customer gets a bundle that has the largest utility to her, or no bundle at all if all utilities are negative. In the single-minded case, every customer who can afford to purchase the requested bundle will do so. Although an envy-free allocation seems fair for all players in the market, companies will definitely not always strive towards this property as it might reduce the revenue. The following example shows that different prices are required if items are available in unlimited and limited supply, and if envy-freeness is required or not.

Example 1.2.1. *Consider a bookstore selling books with titles A, B and C, among others. Four customers currently in the bookstore are interested in a subset of these books. Customer 1 wants to buy book A for at most 10 euro. Customer 2 wants to buy books A and B and spend at most 35 euro on both. Customer 3 is interested in books A and C but only if they cost at most 25 euro. And customer 4 wants to purchase books B and C for at most 30 euro. This situation is schematically displayed in Figure 1.1.*

Let us consider the case where there are sufficiently many books available for all customers. An optimal pricing is to set a price of 10 euro for book A, and 15 euro for each of the other books. Then, customer 1 buys book A for 10 euro, customers 2 and 3 get their requested books for 25 euro, and customer 4 pays 30 euro. All customers can afford to buy the books they like to purchase, and the total revenue for the bookstore is 90 euro. Notice that if the items are available in unlimited supply, any optimal solution is an envy-free solution.

Suppose that there is only one copy of each book left in the bookstore. Then, the seller of the books can decide to set the same prices as before, but

now not all customers are able to purchase the books they are interested in. The optimal revenue of 40 euro is obtained when book A is sold to customer 1 and books B and C are sold to customer 4. Obviously, customers 2 and 3 will not be happy with this allocation because they can afford the books they want, but are not able to buy them. An optimal envy-free solution is to price book A to 15 euro, book B to 20 euro and book C to 11 euro. Then, only customer 2 can afford to buy the books she requests, and the total revenue is 35 euro.

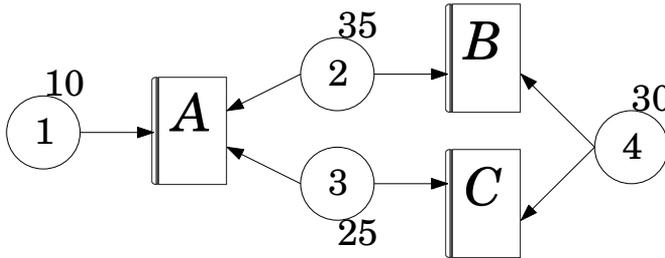


Figure 1.1: Example of the single item pricing problem.

1.2.1 Tollbooth and highway problem

A special case of the single item pricing problem with single-minded customers is the *tollbooth problem*, introduced by Guruswami et al. (2005). The problem is motivated by determining tolls to be charged for segments of a highway system. The capacity on the segments is assumed unlimited. The customers each request to travel several consecutive segments of the highway system. In this setting, we regard the valuation of the customers as a monetary expression of the time saving achieved by traveling the highway as opposed to taking an alternative route. Then, these valuations are naturally public knowledge as is assumed for algorithmic pricing problems. The goal of the owner of the highway is to choose tolls for the segments to maximize the total revenue. We can model this setting such that the highway segments are edges on a simple graph, which represents the complete highway system. Customers request to travel a path in the graph. The problem to the company is to price all edges to maximize the revenue. Similarly as before, the price of a path is equal to the sum of prices for the edges on this path, the customers only travel the path if its price does not exceed the valuation, and the revenue is equal to the total sum of prices for all edges traveled by customers.

An even further restricted problem is the *highway problem*. Here, we determine tolls on the segments of a *single* highway. The underlying graph is thus a path of which customers request subpaths. As this particular problem is extensively discussed in the next three chapters, we present an example to further clarify the problem.

Example 1.2.2. Consider a highway containing three segments, *A*, *B* and *C*. There are several alternative roads next to the highway. They are displayed on the left of Figure 1.2, together with the costs for the travelers for choosing such an alternative road. Suppose there are four customers. The first customer only wants to travel via segment *A*. Based on the alternative road network, her valuation is 7. The second customer wants to travel *B* and *C*. There is no direct alternative connection, so the valuation is $6 + 4 = 10$. The third customer only travels *C* and has valuation 4. The last customer travels all three segments. The cheapest alternative road is to first pass *A* and *B* and afterwards pass *C*, which gives a valuation of $11 + 4 = 15$. The right figure displays a schematic picture of this situation.

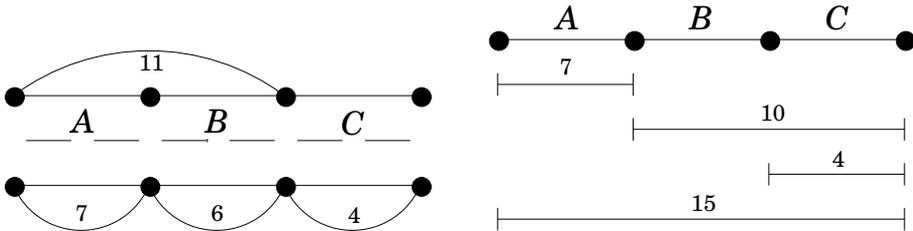


Figure 1.2: Example of the highway problem.

The maximum revenue of 34 can be obtained by charging 5, 6 and 4 at tollbooths *A*, *B* and *C*, respectively. Notice that this same revenue can also be received by setting the prices at the tollbooths to 7, 4, and 4.

1.3 Affine pricing

To introduce the affine pricing problem, we consider the following application. A telephone operator faces the problem to set prices in order to maximize the total revenue. The operator has to determine prices for one minute of calling, sending one text message, establishing a connection, monthly fee, etc. Based on the phone usage of the customers, the operator is well aware of how, how often and during what times the customers use their phones. That

is, the company knows what *contract* a customer requests, and moreover, how much she is willing to pay for it. The contract price is personal to every customer because, next to the prices set by the operator, it includes her personal demand with respect to using her phone. Furthermore, the customer may be charged a fixed cost for accepting the contract. Using the acquired information, the operator can determine all prices to maximize the revenue by solving the *affine pricing problem*. We clarify this problem in an example, and formally define it afterwards.

Example 1.3.1. *In a fixed period of time, four customers are interested in a contract. One customer requests to call 250 minutes and send 25 text messages. Another customer wants to call 100 minutes and send 50 text messages. The third customer requests to call 100 minutes and also send 100 text messages, and the last customer only wants to call 150 minutes. Their valuations are 70, 35, 10 and 45, respectively. The fixed price for accepting the contract is 5 for the first three customers, and 2.50 for the fourth customer.*

The best pricing strategy is to charge 0.25 for one minute of calling and 0.10 per text message. Then, the contract price is $250 \cdot 0.25 + 25 \cdot 0.10 + 5 = 70$ for the first customer, $100 \cdot 0.25 + 50 \cdot 0.10 + 5 = 35$ for the second customer, $50 \cdot 0.25 + 100 \cdot 0.10 + 5 = 27.50$ for the third customer and $150 \cdot 0.25 + 2.50 = 40$ for the fourth customer. The third customer is not able to purchase her requested contract, and consequently, she also does not pay the fixed price. Therefore, the total revenue is 145.

More general, in the affine pricing problem the company wants to price a set of distinct *item types*. These item types can be physical or digital products, but also services, shipping, or the start-up of a process. Every single-minded customer requests one *contract* from the company based on her personal demand on the different item types available. This contract may contain multiple copies of an item type. A customer accepts the contract if the price of the contract does not exceed her *valuation*, which is the maximum amount she is willing to pay for her contract. We might regard this as the amount she has to pay for the same contract at any competing company, which is naturally public knowledge. As a direct consequence hereof, the customer will only accept the contract if it is the cheapest available in the market. However, if the company is able to determine different valuations through market research, this can obviously be used instead. The total price of a customer's contract is equal to a constant price per item type for every unit she requests, and possibly a fixed, customer dependent amount for accepting the contract. In other words, given the prices set by the company, the

price a customer pays for her contract is an *affine function* on her personal request.

Affine pricing can be implemented as a pricing tool in many different companies. Because of the general structure, the item types can be anything from physical product to service, traveling or shipping. When determining the price of a contract, the item types do not have to be comparable. For example, if a customer travels by train, the price of the ticket depends on the distance, the class, the time and day at which she travels, whether she owns a frequent travel card, etc. In this example, the supply can be regarded as unlimited. However, similarly as in single item pricing, the availability of the item types might be limited. Also, the notion of envy-freeness can still be applied in affine pricing.

1.3.1 Price regulation

Using affine pricing to determine the optimal prices for the company, we also touch on the problem in which the company has to deal with price regulations. A price regulation refers to the policy of influencing price strategies by a government agency. Price regulations occur in many industries, for example, environmental or pharmaceutical. In some industries, the government sets price ceilings while in other industries there may be price floors, or the regulation can be imposed through taxation. Reasons for the government to interfere are for example that prices are misleading, there exists price discrimination in a market, companies use predatory pricing to take advantage of customers or price fixing in which several companies make arrangements to artificially keep prices high. The government's main purpose for using price regulation is to protect the customers, without harming the market. A recent example of price regulation is on roaming tariffs in the European Union (European Commission, Information Society, Media Directorate General 2007). *Roaming* is using a mobile phone outside the geographical coverage area of the home network, by means of using a visited network. The price for roaming was on average four times higher than for domestic calls. The repeated warnings to the operators to lower roaming charges voluntarily were not taken seriously. National authorities felt powerless because roaming is international calling. Therefore, the European Commission interfered through price regulation. The goal of the regulation is to have fairer prices charged to customers and among operators, and to improve the transparency of the market.

1.4 Related work

Although pricing is an extensively studied area in economics, algorithmic pricing is a relatively new field of research. Motivated by the availability of customer data, Rusmevichientong (2003) and Rusmevichientong, van Roy, and Glynn (2006) define a pricing problem in which customers are characterized by their valuations for different items and a selection rule describing how a customer selects an item among those she can afford once prices are determined. The problem described is the single item pricing problem with unit-demand customers, as each customer buys at most one item among all items she is interested in. Aggarwal et al. (2004) first considered approximation algorithms and the analysis of the computational complexity for this problem. It is further discussed by Briest and Krysta (2007), Briest (2007), Briest, Hoefer, and Krysta (2008) and Briest (2008).

The single item pricing problem as addressed in this thesis was introduced by Guruswami et al. (2005). They show that the problem is APX-hard, for both limited and unlimited supply. APX is the subclass of NP Optimization problems that admit constant-factor approximation algorithms. APX-hardness denies the existence of a polynomial time approximation scheme (PTAS), which is an algorithm that, for any $\varepsilon > 0$, is guaranteed to find a solution whose revenue is within a factor $(1 + \varepsilon)$ of the optimum revenue, in polynomial time depending on ε . In the same paper, the authors derive an $O(\log n)$ -approximation algorithm for the limited supply problem with unit-demand customers, and an $O(\log n + \log m)$ -approximation for the unlimited supply problem with single-minded customers, where n is the number of customers and m is the number of items. Hartline and Koltun (2005) design near-linear and near-cubic time approximation schemes under the assumption that the number of distinct items is constant. Briest and Krysta (2006) show that the APX-hardness proof from Guruswami et al. (2005) holds for even more restricted instances of the problem. Balcan and Blum (2006) derive an $O(k)$ -approximation for the instance where each customer is interested in bundles containing at most k items, improving the $O(k^2)$ -approximation of Briest and Krysta (2006). For the single item pricing problem with unlimited supply and single-minded customers, Demaine et al. (2006) prove semi-logarithmic inapproximability. More specifically, they show that the problem is hard to approximate within a factor of $O(\log^\varepsilon n)$ for some $\varepsilon > 0$, assuming that $\text{NP} \not\subseteq \text{BPTIME}(2^{O(n^\delta)})$ for all $\delta > 0$. In particular, because of this result it is unlikely that a constant-factor approxima-

tion algorithm exists. There exist two fully polynomial time approximation schemes by Briest and Krysta (2006) and Balcan and Blum (2006) for the problem where the bundles are a *laminar family*, that is, for any two bundles it holds that either one is a subset of the other or they do not intersect. A fully polynomial time approximation scheme (FPTAS) for a maximization problem assures that, for any $\varepsilon > 0$, we have an algorithm that computes a solution with revenue at least $(1 - \varepsilon)$ times the optimum revenue, in time polynomial in the input and $1/\varepsilon$. Finally, Chawla, Hartline, and Kleinberg (2007) describe the single item pricing problem in which the demands of the unit-demand customers are drawn from an explicit probability distribution, in contrast to the assumption that the demands are known to the company. They show that if the valuations are drawn independently for different items, constant approximation guarantees are possible using concepts from the theory of optimal auction design (Myerson 1981).

The tollbooth problem is introduced by Guruswami et al. (2005). This problem is proven to be APX-hard, even if the underlying graph is a star, all valuations are equal to one, and the customers' paths contain at most two edges. The complexity of the highway problem was left open at the introduction by Guruswami et al. (2005). However, as all pricing problems described here are instances of the Stackelberg game, they can be interpreted as a *bilevel linear program* as discussed by e.g. Vicente and Calamai (1994) and Marcotte and Savard (2005). Specifically for the highway problem this means that if either the price vector or the set of winners is known, the problem is polynomially solvable as shown in Section 4.2, even under the requirement of integral prices. Bodlaender and Penninx (2005) and Briest and Krysta (2006) show independently that the seemingly innocent highway problem is NP-hard. Guruswami et al. (2005) propose a polynomial time dynamic programming algorithm when the valuations are bounded by a constant, and a pseudo-polynomial time dynamic programming algorithm when the lengths of the subpaths are bounded by a constant. Balcan and Blum (2006) derive an $O(\log m)$ -approximation algorithm for the highway problem, improving upon the previous $O(\log m + \log n)$ -approximation of Guruswami et al. (2005), where m is the number of highway segments and n is the number of customers. Elbassioni, Sitters, and Zhang (2007) present a quasi-polynomial time approximation scheme (QPTAS) for both the limited and unlimited supply case of the problem. A QPTAS is an approximation scheme whose running time is *quasi*-polynomial. The existence of a QPTAS implies that the problem is not APX-hard, assuming that $NP \not\subseteq$

$\text{DPTIME}[n^{\text{polylog}(n)}]$. Therefore, the existence of a QPTAS can be seen as a strong indication for the existence of a PTAS.

Labbé, Marcotte, and Savard (1998) and Brotcorne et al. (2001) study the problem of setting tolls on a specified subset of edges of a multi-commodity transportation network. This study is different from the tollbooth problem discussed above as the customers travel over the shortest path from their origin to destination via tolled and non-tolled arcs instead of traveling either only tolled edges on a pre-specified path or only non-tolled edges. A similar problem, but focussed to setting tolls on a single highway, is discussed by Heilporn et al. (2007). For further reading on Stackelberg pricing in networks, we refer to Dewez (2004), van der Kraaij (2004), Roch, Savard, and Marcotte (2005), van Hoesel (2006) and Bouthou et al. (2007).

The affine pricing technique in algorithmic pricing is introduced by Grigoriev et al. (2007). The difference between affine pricing and single item pricing is that in affine pricing customers can request multiple copies of one item type and the price of the contract is personal. Consider again the application of the telephone operator, and assume that the operator wants to determine a price per minute calling. For every minute, the customer will be charged this same price. If we would solve the same problem with single item pricing, then we have to price every single minute separately. The demand of every customer is similar as in the affine pricing problem, which means that if a customer wants to call for x minutes at various time periods, then she requests one copy of the item that models calling in a particular minute, and her bundle contains x items. The problem is then to determine a price for every single minute. Thus, the affine pricing problem is in fact a generalization of the single item pricing problem because we can regard every single copy of all item types in the affine pricing problem as separate items in the single item pricing problem.

1.5 Outline

The *single item pricing* problem with single-minded customers is discussed in the first part of this thesis. In Chapter 2 we consider this problem with unlimited supply. We impose a natural monotonicity constraint enforcing that larger bundles are at least as expensive as smaller ones. We show that this problem is strongly NP-hard and derive a polynomial time approximation scheme, thereby breaking the semi-logarithmic inapproximability result by Demaine et al. (2006). As a special case we also address the highway

problem. We show that this problem remains weakly NP-hard under the monotonicity condition, and we derive a simple approximation algorithm that is logarithmic in the largest valuation.

In Chapter 3, we address the single item pricing problem with unlimited supply and we show that the problem is trivially solvable if all customers have the same average valuation, that is, the valuation divided by the size of the requested bundle. But, even for the slightest deviation in average valuation, the problem is strongly NP-hard immediately. These results remain true also if we pose monotonicity constraints on the valuations and average valuations. That is, customers requesting a large bundle should have a higher valuation but lower average valuation than customers interested in a smaller bundle. Moreover, we present an approximation algorithm with a performance guarantee that is logarithmic in the largest deviation between any two customers' average valuations. In the last part of this chapter, we study the highway problem. The approximation algorithm we create for the pricing problem with arbitrary bundles carries over to the highway problem. Also, this problem is trivially solvable if the average valuations are equal for all customers. Even the slightest deviation makes that this problem becomes NP-hard.

We describe the algorithmic complexity of the single item pricing problem in capacitated, undirected networks in Chapter 4. The edges of an underlying graph are regarded as the items for sale and the customers' bundles are paths in the graph. If the graph is a path (highway problem) and capacity is unlimited then we derive an approximation algorithm that is logarithmic in the largest valuation. The problem with a constant upper bound on the limited edge capacities can be solved using a dynamic programming algorithm. Based on this dynamic program, we derive a fully polynomial time approximation scheme. The problem with limited edge capacities and with a cycle as underlying graph is NP-hard, and can be solved by an adjusted dynamic program in pseudo-polynomial time. Finally, with unit edge capacity, the pricing problem is polynomially solvable if the underlying graph is a path, cycle or tree. The same problem on a grid is surprisingly much harder.

The second part of this thesis focuses on *affine pricing*. In Chapter 5 we present an algorithm to solve the affine pricing problem in polynomial time, if the number of distinct item types is constant. This result holds for the envy-free problem with unlimited supply of item types, both for single-minded and unit-demand customers, and we derive a near-optimal solution if the supply is limited. Moreover, we show that the affine pricing problem

is a generalization of the single item pricing problem, which means that for a non-constant number of item types available in unlimited supply, the hardness results known for single item pricing also hold for affine pricing. Moreover, if the availability of item types is limited, the problem is hard to approximate within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$, where n is the number of customers. After the theoretical results, we apply the affine pricing algorithm in a practical study, where the number of item types is constant, the customers are single-minded and the supply is unlimited. As the running time of the algorithm is large for huge instances, we present a local search algorithm that gives promising results on the tested instances.

In Chapter 6 we consider the situation in which a company wishes to implement a price vector different from the current price vector. It is not always feasible to implement new prices instantly, because of the involvement of public authorities, the strategy towards competitors or the customers' perception of the company. This forces the company to implement the new prices gradually. We impose the restriction that for the customers that accept their contract in the optimal pricing, the contract price never increases more than some predetermined percentage per time period. We determine the minimum number of time periods necessary to implement the new prices under this assumption. Furthermore, we describe how to maximize the total revenue over time if a maximum number of time periods is given in which the new pricing must be implemented.

Motivated by the current price regulation on roaming tariffs, we closely look into the problem where the pricing strategy of the company is regulated by a government in Chapter 7. We model the setting as a three-level program, and present a polynomial time algorithm to solve this problem. Furthermore, we analyze the current policy of the European Commission, and compare different price regulative policies. We test our theoretical results on a data set.

1.6 List of publications

This thesis is based on the following publications:

- A. Grigoriev, J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld (2008). Optimal bundle pricing with monotonicity constraint. *Operations Research Letters* 36, 609–614.
A preliminary version of this work appeared as:
A. Grigoriev, J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld (2007). Bundle pricing with comparable items. In L. Arge, M. Hoffmann, and E. Welzl (Eds.), *Algorithms - ESA 2007*, Volume 4698 of *Lecture Notes in Computer Science*, pp. 475–486. Springer.
- A. Grigoriev, J. van Loon, R. Sitters, and M. Uetz (2009). Optimal pricing of capacitated networks. *Networks* 53(1), 79–87.
A preliminary version of this work appeared as:
A. Grigoriev, J. van Loon, R. Sitters, and M. Uetz (2006). How to sell a graph: Guidelines for graph retailers. In F. Fomin (Ed.), *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, Volume 4271 of *Lecture Notes in Computer Science*, pp. 125–136. Springer.
- A. Grigoriev, J. van Loon, and M. Uetz (2008). Algorithms for optimal price regulations. In C. Papadimitriou and S. Zhang (Eds.), *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, Volume 5385 of *Lecture Notes in Computer Science*, pp. 362–373. Springer.
- A. Grigoriev, J. van Loon, and M. Uetz (2008). On the complexity of the highway pricing problem. METEOR Research Memorandum RM/08/030, Universiteit Maastricht.
- A. Berger, A. Grigoriev, and J. van Loon (2008). Price strategy implementation. METEOR Research Memorandum RM/08/035, Universiteit Maastricht.

Part I

Single Item Pricing

Chapter 2

Monotonicity Constraint on Prices

In this chapter we consider the single item pricing problem with unlimited supply. We know that the problem is inapproximable within a semi-logarithmic factor in the number of customers by Demaine et al. (2006). Therefore, we suggest a natural monotonicity constraint, enforcing that the price of larger bundles is at least as high as the price of smaller ones. We show that this problem remains strongly NP-hard, and we derive a PTAS. Finally, we also discuss the highway problem with monotonicity constraint.¹

2.1 Introduction

We consider the situation where a company wants to sell a set of (digital) items $I = \{1, \dots, m\}$ to a set of single-minded customers $J = \{1, \dots, n\}$. Every item $i \in I$ is available in unlimited supply. Each customer $j \in J$ is interested in exactly one subset of items $I_j \subseteq I$, her bundle, and has a positive valuation b_j for this bundle, which is the maximum amount she is willing to pay. Without loss of generality we assume that $b_j \geq 1$ for $j \in J$. Let B be the highest valuation, that is, $B = \max_{j \in J} \{b_j\}$. Let $p_i \geq 0$ be the price of item $i \in I$. The price of a bundle I_j is therefore $\sum_{i \in I_j} p_i$. A customer is a *winner* if the total price of her bundle does not exceed her valuation, and a *loser* otherwise. The set of winners is denoted by $W \subseteq J$.

¹This chapter is based on Grigoriev, van Loon, Sviridenko, Uetz, and Vredeveld (2007) and Grigoriev, van Loon, Sviridenko, Uetz, and Vredeveld (2008).

Definition 2.1.1. A feasible solution consists of a positive price p_i for every item $i \in I$ and a set of winners $W \subseteq J$ such that all winners can afford their bundle, that is, $\sum_{i \in I_j} p_i \leq b_j$ for all $j \in W$, and the solution is envy-free, that is, $\sum_{i \in I_j} p_i > b_j$ for all $j \in J \setminus W$. The objective is to maximize the company's total revenue $\Pi = \sum_{j \in W} \sum_{i \in I_j} p_i$.

Notice that any solution will be trivially *envy-free*, meaning that a customer not receiving her bundle cannot afford it. This is an artifact of the unlimited supply of items.

In a sequence of recent papers, several algorithms and complexity results have been derived for such price optimization problems; see e.g. Aggarwal et al. (2004), Guruswami et al. (2005), Hartline and Koltun (2005), Balcan and Blum (2006), Briest and Krysta (2006), Elbassioni, Sitters, and Zhang (2007). Many of the positive results achieve only logarithmic approximation guarantees in the number of customers, the number of items or both, and often somewhat trivial algorithms suffice to achieve such bounds, e.g. uniform prices for all items. Demaine et al. (2006) delivered a reason for this observation on the performance of algorithms, showing that the problem does not allow for semi-logarithmic approximation algorithms in general. In this chapter, we introduce a very natural *monotonicity* constraint on bundle prices that allows us to derive results that break this semi-logarithmic inapproximability barrier. More specifically, we impose the condition that the price of any bundle of size k must not exceed the price of a bundle of size $k + 1$ or larger, for any k . That is, the monotonicity constraint that we impose on the set of feasible price vectors is

$$\sum_{i \in I'} p_i \leq \sum_{i \in I''} p_i \text{ whenever } |I'| < |I''|, \quad (2.1)$$

for any two subsets of items I' and I'' . The condition has a meaningful economic interpretation in settings where items are different, yet comparable in the sense that their prices cannot differ too much. It only requires that larger bundles are at least as expensive as smaller ones. Notice that we do not require a monotonicity condition on bundles of the same size, as this would immediately yield that all items must receive the same price. Yet, the condition is quite strong, as it requires that most item prices are of the same order of magnitude.

We show in Section 2.2 that the single item pricing problem is strongly NP-hard even with the monotonicity constraint on bundle prices as defined in (2.1). Moreover, we derive a polynomial time approximation scheme for

that problem, with a time complexity of $O(nm^{8/\varepsilon}(\log B)^{8/\varepsilon})$. The PTAS crucially exploits the fact that the monotonicity condition (2.1) yields most item prices to be of the same order of magnitude. In Section 2.3 we address the highway problem, where items can be seen as edges on a simple path and customers request a subpath. The monotonicity constraint is only imposed on subpaths. We show that this problem remains weakly NP-hard under the monotonicity constraint, and we derive an $O(\ln B)$ -approximation algorithm.

2.2 Arbitrary bundles

We settle the computational complexity of the single item pricing problem in which customers requests arbitrary bundles and the prices satisfy the monotonicity constraint, defined in (2.1). First, we prove that this particular problem is strongly NP-hard. Then, we derive a polynomial time approximation scheme.

2.2.1 Complexity

We show that the pricing problem with monotonicity constraint is strongly NP-hard by using a reduction from INDEPENDENTSET, which is known to be NP-hard (Garey and Johnson 1979) and defined as follows. Given a graph $G = (V, E)$ and integer $s \leq |V|$, does there exist a set of vertices $V' \subseteq V$ such that no two vertices in V' are joined by an edge in E , with cardinality $|V'| \geq s$?

We define an instance \mathcal{I} of the pricing problem in the following way. Let M be an integer such that $M \geq 2|E|^2 + 4|E| + 2$. For every vertex $v \in V$ we create a *vertex-item*, and for every edge $e \in E$ we introduce an *edge-item*, that is, $I = V \cup E$. For every item $i \in I$, there are $M + 2$ customers that request the bundle consisting of only this item. One of these customers has valuation M , and all other customers have valuation $M + 1$. Moreover, for every edge $e = \{u, v\} \in E$, there are four more customers. One customer requests bundle $\{u, e\}$, one requests bundle $\{v, e\}$, and two customers request bundle $\{u, v\}$. These four customers each have valuation $2M + 1$.

Figure 2.1 displays how an edge $e = \{u, v\} \in E$ of graph G is represented in instance \mathcal{I} . The bundle of *one* customer is represented by a dashed line, and this customer has valuation M for that particular bundle. Every solid line represents a bundle of $M + 1$ customers who have valuation $M + 1$. Each dotted line represents a bundle of *one* customer who has valuation $2M + 1$.

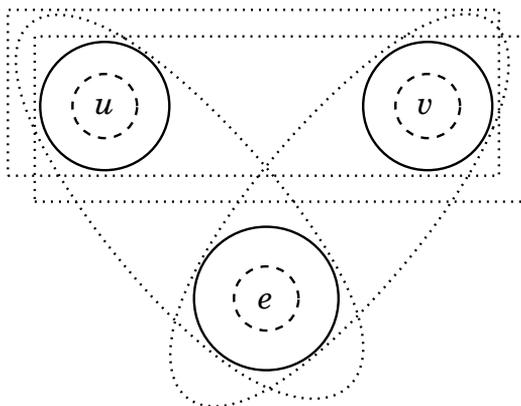


Figure 2.1: Instance \mathcal{S} for edge $e = \{u, v\} \in E$ from the reduction of INDEPENDENT-SET.

We prove strong NP-hardness of the single item pricing problem with monotonicity constraint in Theorem 2.2.5. In this proof, we need the property that in an optimal solution to the above created instance \mathcal{S} of the pricing problem with monotonicity constraint all prices are either M or $M + 1$. Therefore, we first collect some preliminary observations.

Lemma 2.2.1. *In the optimal solution to the pricing problem on \mathcal{S} , all prices are within the interval $[M, M + 1]$.*

Proof. Assume the claim is not true, and let p be an optimal solution that violates it. If all items have prices $p_i \leq M + 1$ and there exists at least one item with price $p_i < M$, then it can easily be verified that the price vector p' , defined by $p'_i = \max\{p_i, M\}$, yields the same set of winners and the revenue has increased compared to p . Hence, p cannot be optimal.

Therefore, assume that at least one item, h , has a price $p_h > M + 1$ and consider the price vector p' defined by $p'_i = p_i$ for $i \neq h$ and $p'_h = M + 1$. Since item h belongs to at most $3|E|$ bundles containing two items, the decrease in revenue of these bundles due to the price change is at most $3|E|M$. This is because, without loss of generality, the price of an item is at most $2M + 1$ in an optimal solution, that is, at most equal to the largest valuation. On the other hand, $M + 1$ customers can now afford bundle $\{h\}$. Therefore, the increase in revenue due to these customers is $(M + 1)^2$. Thus, the total revenue increases by at least $M^2 + (2 - 3|E|)M + 1$ which is positive since $M > 2|E|^2$. Hence, the price vector p cannot be optimal. \square

Lemma 2.2.2. *In an optimal solution to the pricing problem on \mathcal{I} , all customers requesting a bundle that contains two items are winners.*

Proof. Suppose that there exists a bundle containing two items, say $I_j = \{i_1, i_2\}$, that is priced higher than $2M + 1$ and let the prices of the two items be $p_{i_1} = M + d_1$ and $p_{i_2} = M + d_2$. Combining this with the result of the previous lemma, we know that $0 < d_1, d_2 \leq 1$. As each item is in at most $3|E|$ bundles of size two, decreasing the price of item i_1 to M increases the total revenue by at least

$$-((M + 1) + (3|E| - 1))d_1 + M + 2M + d_2 \geq 2M - 3|E|,$$

which is positive since $M > 3|E|/2$. \square

Lemma 2.2.3. *In an optimal solution to the pricing problem on \mathcal{I} , all prices are integral.*

Proof. By Lemma 2.2.1, we know that each item has a price $p_i = M + d_i$, for $0 \leq d_i \leq 1$. Suppose that there exists an item i with $0 < d_i < 1$ and that this item corresponds to an edge $e = \{u, v\}$. If both $d_u = d_v = 0$, then we can change the price of the edge-item e to $p'_e = M + 1$ without affecting the set of winners and increasing the total revenue. Therefore, assume that $d_u > 0$ and, without loss of generality, $d_u \geq d_v$. By Lemma 2.2.2, we know that $d_u + d_v \leq 1$ and thus $d_v \leq 1/2$. Consider the prices p' defined by $p'_e = M + 1$, $p'_u = p'_v = M$, and $p'_i = p_i$ for all other items i . The increase in revenue due to this price change for the bundles containing two items corresponding to the edge $e = \{u, v\}$ is $1 - (d_u + d_e) + 1 - (d_v + d_e) - 2(d_u + d_v)$, whereas the total decrease in revenue for all other bundles containing two items is bounded by $3(|E| - 1)(d_u + d_v)$. The increase in revenue caused by the bundles $\{e\}$ and $\{u\}$ is $(1 - d_e)(M + 1) + M - d_u(M + 1)$, whereas the increase in revenue for bundle $\{v\}$ is $M - (M + 1)d_v \geq 0$ if $d_v > 0$ and 0 otherwise. Hence, the total increase in revenue due to this price change is at least

$$\begin{aligned} & 1 - (d_u + d_e) + 1 - (d_v + d_e) - 2(d_u + d_v) - 3(|E| - 1)(d_u + d_v) \\ & + (1 - d_e)(M + 1) + M - d_u(M + 1) \\ \geq & M - (3|E| - 1)(d_u + d_v) + (1 - d_e - d_u)(M + 1) \geq M - (3|E| - 1) > 0, \end{aligned}$$

as $M > 3|E| - 1$. Hence, any solution in which an edge-item e has a fractional price cannot be optimal.

On the other hand, suppose that all edge-items $e \in E$ have price $p_e \in \{M, M + 1\}$ and there exists a vertex-item u with $M < p_u < M + 1$. Then by

Lemma 2.2.2 we know that there exists an item u such that all its neighbors v with $\{u, v\} \in E$ have price $p_v \leq M+1/2$. Moreover, we may assume that there exists a neighbor v such that $p_v > M$ as otherwise we may set $p_u = M+1$ without affecting the set of winners, but increasing the total revenue. Let $d = \max\{d_v : \{u, v\} \in E\}$, then $0 < d \leq 1/2$. Consider the prices p' defined by $p'_u = M+1$, $p'_v = M$ for v such that $\{u, v\} \in E$ and $p'_i = p_i$ for all other items i . Then the revenue for each bundle containing two items is decreased by at most $2d$ and as there are in total $4|E|$ of these bundles the total decrease in revenue due to this price change is bounded by $4|E|(2d) \leq 4|E|$. As there was at least one neighbor of u that had a price $p_v > M$, the change in prices leads to an increase of revenue of at least

$$(1 - d_u)(M + 1) + M - d(M + 1) \geq (1 - d_u - d)(M + 1) + M \geq M,$$

as $d_u + d \leq 1$ by Lemma 2.2.2. Hence the total increase in revenue due to this price change is at least $M - 4|E| > 0$ as $M > 4|E|$. Hence, no solution with at least one fractional price can be optimal. \square

Lemma 2.2.4. *An optimal solution to instance \mathcal{I} satisfies the monotonicity constraint.*

Proof. We know that p_i is either M or $M+1$ for all $i \in I$ by Lemma 2.2.1 and Lemma 2.2.3. Consequently, for any two set $I', I'' \subseteq I$ for which holds that $|I'| < |I''|$ we have

$$\sum_{i \in I'} p_i \leq |I'|(M + 1) < (|I'| + 1)M \leq |I''|M \leq \sum_{i \in I''} p_i,$$

where the second inequality holds because $M \geq 2|E|^2 + 4|E| + 2 > |V| + |E| = |I| > |I'|$. \square

We use the above four lemmas to prove that the single item pricing problem with arbitrary bundles satisfying the monotonicity constraint is strongly NP-hard.

Theorem 2.2.5. *The single item pricing problem with monotonicity constraint is strongly NP-hard.*

Proof. Consider instance \mathcal{I} . We claim that there exists an independent set of size s in G if and only if there is a solution to the single item pricing problem with revenue at least $f(s) = (M^2 + 2M)|V| + (M^2 + 10M + 3)|E| + s$.

Given an independent set $V' \subseteq V$ with $|V'| \geq s$. Let $E_0 = \{e = \{u, v\} \in E : u, v \notin V'\}$. Let $p_v = M + 1$ if $v \in V'$ and M otherwise. Let $p_e = M + 1$ if $e \in E_0$ and M otherwise. By Lemma 2.2.4 we know that this pricing satisfies the monotonicity constraint. The revenue of this solution is

$$\begin{aligned} \Pi &= (M + 1)(M|V| + |V'| + M|E| + |E_0|) + M|V \setminus V'| + M|E \setminus E_0| \\ &\quad + 8M|E| + 2|E_0| + 3|E \setminus E_0| \\ &= (M^2 + 2M)|V| + (M^2 + 10M + 3)|E| + |V'|, \end{aligned}$$

which is greater than or equal to $f(s)$.

For the converse, assume there is an optimal solution to the pricing problem that satisfies the monotonicity constraint with revenue at least $f(s)$. We know by Lemma 2.2.1 and Lemma 2.2.3, that the price for any item is either M or $M + 1$. Let $V' = \{v \in V : p_v = M + 1\}$, then V' is an independent set in G , as by Lemma 2.2.2 we know that for any $e = \{u, v\} \in E$ with $u \in V'$, $p_v \leq M$. Define $E_0 = \{e = \{u, v\} \in E : p_e = M + 1\}$. Then for any $e = \{u, v\} \in E_0$, we have that $p_u = p_v \leq M$ by Lemma 2.2.2. The revenue for the four bundles containing two items corresponding to an edge $e \in E \setminus E_0$ is bounded by $3(2M + 1) + 2M = 8M + 3$. The total revenue is bounded from above by

$$\begin{aligned} &|V'|(M + 1)^2 + |V \setminus V'| (M^2 + 2M) + |E_0|(M + 1)^2 + |E \setminus E_0|(M^2 + 2M) \\ &+ |E_0|(2(2M + 1) + 2(2M)) + |E \setminus E_0|(8M + 3) \\ &= (M^2 + 2M)|V| + |V'| + (M^2 + 10M + 3)|E|. \end{aligned}$$

As the total revenue is at least $f(s)$, we conclude that $|V'| \geq s$. \square

2.2.2 Approximation scheme

In this section we show that the pricing problem with monotonicity constraint admits a polynomial time approximation scheme (PTAS). A PTAS is an algorithm which takes an instance of an optimization problem and, for any $\varepsilon > 0$, produces a solution that is within a factor $(1 + \varepsilon)$ of being optimal, in polynomial time depending on ε .

Assume, without loss of generality, that $p_1 \leq p_2 \leq \dots \leq p_m$. Then by the monotonicity constraint, we know $2p_2 \geq p_1 + p_2 \geq p_m$. Similarly, $3p_3 \geq p_1 + p_2 + p_3 \geq p_{m-1} + p_m \geq 2p_{m-1}$, etc.

Lemma 2.2.6. *Suppose $p_1 \leq p_2 \leq \dots \leq p_m$. Any pricing of the items satisfying the monotonicity constraint also satisfies*

$$k p_k \geq (k - 1)p_{m-k+2}, \quad k = 2, \dots, \left\lceil \frac{m}{2} \right\rceil. \quad (2.2)$$

The idea for the PTAS is now the following. First, we restrict the prices to powers of $(1 + \delta)$ for some $\delta > 0$. Then, Lemma 2.2.6 says that except for a constant number of the cheapest and most expensive items, all items have prices in roughly the same range. Therefore we can price all except a constant number of items uniformly with the same price, without losing too much in terms of the total revenue. We therefore enumerate over all possible uniform prices for the bulk of the items, and over all possible combinations of prices for the remaining (constant number of) items.

Theorem 2.2.7. *The pricing problem with monotonicity constraint admits a PTAS. The time complexity is $O(nm^{8/\varepsilon}(\log B)^{8/\varepsilon})$, where $\varepsilon > 0$ is the precision of the PTAS.*

Proof. Given an instance of the pricing problem and an $\varepsilon > 0$, let $\delta = \varepsilon/4$, and for convenience assume that $1/\delta$ is integral. Let $p_1 \leq \dots \leq p_m$ be the prices in an optimal solution, satisfying the monotonicity constraints. Define the subsets of items $S = \{i \in I : i \leq \frac{1}{\delta}\}$, $M = \{i \in I : 1 + \frac{1}{\delta} \leq i \leq m + 1 - \frac{1}{\delta}\}$ and $L = \{i \in I : i \geq m + 2 - \frac{1}{\delta}\}$, as shown in Figure 2.2. Note that $M = \emptyset$ if $\varepsilon \leq 8/(m + 1)$, in which case the number of items is in $O(1/\varepsilon)$.

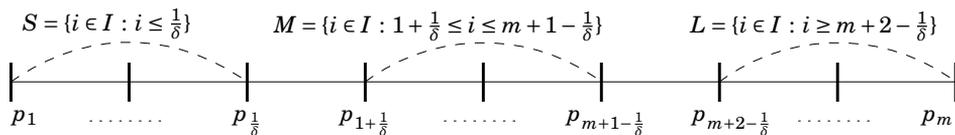


Figure 2.2: Subsets of items S , M and L .

We round down the prices of all items in S and L to powers of $(1 + \delta)$. Moreover, we price all items in M uniformly at price $p_{1+1/\delta}$, rounded down to a power of $(1 + \delta)$. Let us call the new prices p' , and let us call p'_M the price of items in M . First observe that the order of prices does not change. We next argue that we do not lose too much by this rounding. Clearly, since we round down, the set of winners can only increase. Moreover, we lose at most a factor $(1 + \delta)$ on items in S and L . Finally, consider the items in M . By inequality (2.2), we have

$$\left(1 + \frac{1}{\delta}\right)p_{1+1/\delta} \geq \frac{1}{\delta}p_{m+1-1/\delta} \Leftrightarrow (1 + \delta)p_{1+1/\delta} \geq p_{m+1-1/\delta}.$$

In other words, the price for the most expensive item in M differs from the cheapest item in M by a factor of at most $(1 + \delta)$. Hence, on items in M we lose a factor of at most $(1 + \delta)^2$.

Now we have a structured solution, but it may violate the monotonicity constraint. In case of a violation, we divide the highest prices by a factor of $(1 + \delta)$, so that the new price vector satisfies the monotonicity constraint. Note that in this case, we loose at most a factor of $(1 + \delta)^3$ compared to the optimal pricing vector. The question remains is which prices need to be divided by $(1 + \delta)$. To this end, we define $s = \max\{0, i : p'_i < p'_M\}$ as the number of small prices and $\ell = \max\{0, i : p'_{m+1-i} > p'_M\}$ as the number of high prices. If $s > \ell$, then the new prices are defined by $p''_i = p'_i$ for $1 \leq i \leq s$ and for $s + 1 \leq i \leq m$ by $p''_i = p'_i/(1 + \delta)$. Otherwise, if $s \leq \ell$, then the new prices are defined by $p''_i = p'_i$ for $1 \leq i \leq m - \ell$ and for $m - \ell + 1 \leq i \leq m$ by $p''_i = p'_i/(1 + \delta)$. Let $k = \max\{s - 1, \ell\}$. Then, if there is a violation of the monotonicity constraint in the new pricing, sets $\{1, \dots, k + 1\}$ and $\{m + 1 - k, \dots, m\}$ are the two most violating sets, as $p''_{k+2} = p''_{m-k} = p'_M$. Therefore, we only need to verify that for these two sets the monotonicity constraint holds true. First notice that by the way of defining k and the prices p'' , we have that $p''_i = p'_i$ for $1 \leq i \leq k + 1$ and $p''_i = p'_i/(1 + \delta)$, for $m + 1 - k \leq i \leq m$. Hence, we have that

$$\sum_{i=1}^{k+1} p''_i = \sum_{i=1}^{k+1} p'_i \geq \sum_{i=1}^{k+1} \frac{p_i}{1 + \delta} \geq \sum_{i=m+1-k}^m \frac{p_i}{1 + \delta} \geq \sum_{i=m+1-k}^m \frac{p'_i}{1 + \delta} = \sum_{i=m+1-k}^m p''_i,$$

where the second inequality is due to the monotonicity of the original set of prices p . From this it follows that the structured solution p'' satisfies the monotonicity constraint.

The PTAS now consists of enumerating all possible structured solutions, which is sufficient to obtain a feasible solution that differs from the optimal solution by a factor of at most $(1 + \delta)^3 < (1 + \epsilon)$. There are $\binom{m}{-1+2/\delta}$ possible choices for $S \cup L$. Since all prices are powers of $(1 + \delta)$, there are $\log B$ possible prices. Given that all items in M have the same price, there are at most $(\log B)^{2/\delta}$ structured solutions for each choice of $S \cup L$. Computation of the revenue for any such solution takes $O(nm)$ time. This together with $\delta = \epsilon/4$ yields the claimed time complexity, where the constant hidden in the O -notation depends on ϵ . \square

2.3 Highway problem

A particularly intriguing special case of the general pricing problem considered so far is the *highway problem* as introduced by Guruswami et al.

(2005). Let $I = \{1, \dots, n\}$ represent the highway segments, which we can regard as consecutive edges on a simple path. Every customer $j \in J$ requests a *subpath* of the highway, denoted by $I_j \subseteq I$, and we assume that each I_j is of the form $I_j = \{k, k + 1, \dots, \ell\}$, where $1 \leq k \leq \ell \leq m$. We may assume that the valuation b_j of customer $j \in J$ for subpath I_j is a monetary expression for the time saving that can be realized by using the highway instead of the next-fastest alternative route. Similarly as in the single item pricing problem with arbitrary bundles, the company determines prices $p_i \geq 0$ for the items $i \in I$. The price of bundle I_j is $\sum_{i \in I_j} p_i$. A customer is a winner if the total price of her bundle is not more than her valuation, and a loser otherwise. The goal of the company is to maximize the revenue $\Pi = \sum_{j \in W} \sum_{i \in I_j} p_i$, where $W \subseteq J$ is the set of winners.

In the highway problem, it is most natural to assume that the monotonicity constraint holds for any two *subpaths* only, but not necessarily for arbitrary subsets of items. Because of the consecutiveness of the items, the strong NP-hardness result and the PTAS for the single item pricing problem with monotonicity constraint and arbitrary bundles, established in the previous section, do not carry over to the highway problem with price monotonicity on subpaths. Also notice that the weak NP-hardness for the general highway problem by Briest and Krysta (2006) does not automatically yield weak NP-hardness for the problem with price monotonicity, as optimal prices in that completeness proof are not monotone in the length of the subpaths. We next derive weak NP-hardness for the problem with monotonicity constraint, and we present an $O(\ln B)$ -approximation algorithm.

2.3.1 Complexity

We prove NP-completeness of the highway problem with monotonicity constraint by a reduction from EQUALCARDINALITYSUBSETSUM defined in the following way.

Definition 2.3.1 (EQUALCARDINALITYSUBSETSUM). *Given a set of positive integers a_1, a_2, \dots, a_{2L} and nonnegative integer A such that $a_1 \leq a_2 \leq \dots, \leq a_{2L}$, $a_{2L} > \sum_{\ell=1}^{2L-1} a_\ell$ and $0 < A < a_{2L}$. Does there exist a set $S \subseteq \{1, \dots, 2L\}$ such that $\sum_{\ell \in S} a_\ell = A$ and $|S| = L$?*

In Theorem 2.3.2, we show that this problem is NP-complete by a reduction from SUBSETSUM; see Garey and Johnson (1979). This problem is defined as follows. Given a set of positive integers x_1, x_2, \dots, x_K and nonnegative integer X . Does there exist a set $T \subseteq \{1, \dots, K\}$ such that $\sum_{k \in T} x_k = X$?

Theorem 2.3.2. EQUALCARDINALITYSUBSETSUM is NP-complete.

Proof. Consider an instance of the NP-complete problem SUBSETSUM. Assume without loss of generality that $x_1 \leq x_2 \leq \dots \leq x_K$. We create an instance of EQUALCARDINALITYSUBSETSUM as follows: let $L = K + 1$, $A = X + \sum_{k=1}^K x_k$, $a_\ell = 0$ for all $\ell \in \{1, \dots, K = L - 1\}$, $a_\ell = x_{\ell-K}$ for all $\ell \in \{K + 1 = L, \dots, 2K = 2L - 2\}$, $a_{2L-1} = \sum_{k=1}^K x_k$ and $a_{2L} = 2\sum_{k=1}^K x_k + 1$. This instance satisfies the properties of EQUALCARDINALITYSUBSETSUM in Definition 2.3.1. We claim that there exists a solution to SUBSETSUM if and only if there exists a solution to EQUALCARDINALITYSUBSETSUM.

Given a set $T \subseteq \{1, \dots, K\}$ such that $\sum_{k \in T} x_k = X$. Define $S = \{K + k : k \in T\} \cup \{1, \dots, L - 1 - |T|\} \cup \{2L - 1\}$. Then,

$$\sum_{\ell \in S} a_\ell = \sum_{k \in T} x_k + \sum_{\ell=1}^{L-1-|T|} a_\ell + a_{2L-1} = X + 0 + \sum_{k=1}^K x_k = A,$$

and $|S| = |T| + (L - 1 - |T|) + 1 = L$. For the converse, given set S such that $\sum_{\ell \in S} a_\ell = A$ and $|S| = L$. First, note that $2L \notin S$ as $a_{2L} = 2\sum_{k=1}^K x_k + 1 > A$ and $2L - 1 \in S$ as $\sum_{\ell=1}^{2L-2} a_\ell = \sum_{k=1}^K x_k < A$. Define $T = S \cap \{L, \dots, 2L - 2\}$, with

$$\sum_{k \in T} x_k = \sum_{\ell \in S} a_\ell - \sum_{\ell \in S \cap \{1, \dots, L-1\}} a_\ell - a_{2L-1} = \left(X + \sum_{k=1}^K x_k \right) - 0 - \sum_{k=1}^K x_k = X.$$

Thus T is a solution to SUBSETSUM. \square

Consider an instance of EQUALCARDINALITYSUBSETSUM as in Definition 2.3.1. We now create an instance \mathcal{H} of the highway problem as follows. For every $\ell = 1, \dots, 2L$, we define $a'_\ell = a_\ell + a_{2L}$, and we introduce a *gadget*. Every gadget ℓ consists of four items $i_{\ell,1}, i_{\ell,2}, i_{\ell,3}, i_{\ell,4}$. Furthermore, there are $4 + 8L$ customers in every gadget ℓ , one customer requests bundle $\{i_{\ell,1}\}$ with valuation $2M - \frac{1}{2}a'_\ell$, one customer requests bundle $\{i_{\ell,2}\}$ with valuation a'_ℓ , $8L$ customers request bundle $\{i_{\ell,2}, i_{\ell,3}\}$ with valuation $2M + (\frac{1}{2} + \frac{1}{8L})a'_\ell$, one customer requests $\{i_{\ell,3}\}$ with valuation $2M - \frac{1}{2}a'_\ell$, and one customer requests bundle $\{i_{\ell,4}\}$ with valuation M , where M is a sufficiently large integer. For gadget $2L$, there is one additional customer who is interested in bundle $\{i_{2L,1}, i_{2L,2}, i_{2L,3}, i_{2L,4}\}$ with valuation $5M$. Finally, there is one *big* customer requesting all items with valuation $10ML + \frac{1}{8}a_{2L} + \frac{1}{8L}A$. Thus, the instance of the highway pricing problem has $2L(4 + 8L) + 2$ customers and $8L$ items, and is displayed in Figure 2.3.

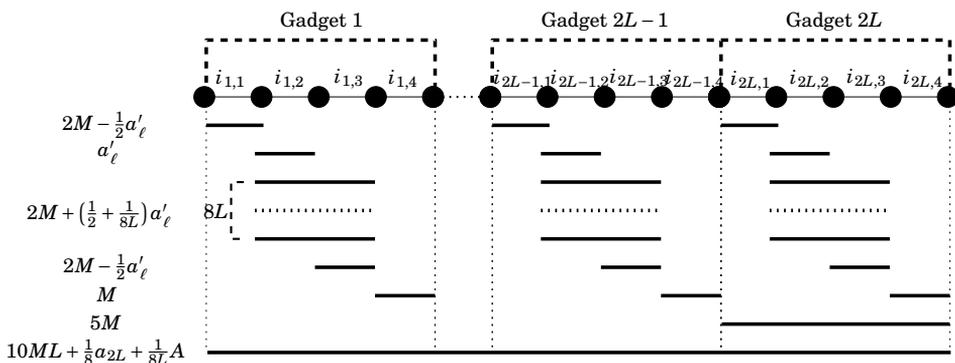


Figure 2.3: Instance \mathcal{H} of the highway problem.

To prove NP-hardness, we claim that there exists a set $S \subseteq \{1, \dots, 2L\}$ such that $\sum_{\ell \in S} a_\ell = A$ and $|S| = L$ if and only if there is a feasible solution to the highway problem, satisfying the monotonicity constraint, with a total revenue $\Pi \geq (20 + 32L)ML + 5M + 4L \sum_{\ell=1}^{2L} a_\ell + (8L^2 + \frac{1}{8})a_{2L} + \frac{1}{8L}A$. In the NP-hardness proof of Theorem 2.3.5, we use for every gadget one out of two following price vectors.

$$p_\ell^1 = \left(2M - \frac{1}{2}a'_\ell, a'_\ell, 2M - \frac{1}{2}a'_\ell, M \right) \quad (2.3)$$

$$p_\ell^2 = \left(2M - \frac{1}{2}a'_\ell, (1 + \frac{1}{8L})a'_\ell, 2M - \frac{1}{2}a'_\ell, M \right). \quad (2.4)$$

To prove the necessary condition of the claim, we show in Lemma 2.3.3 that each gadget will be priced at p_ℓ^1 or p_ℓ^2 .

Lemma 2.3.3. *Suppose that there exists a solution to instance \mathcal{H} of the highway problem with revenue at least $(20 + 32L)ML + 5M + 4L \sum_{\ell=1}^{2L} a_\ell + (8L^2 + \frac{1}{8})a_{2L} + \frac{1}{8L}A$. Then, the items in gadget $\ell \in \{1, \dots, 2L\}$ are priced according to either price vector p_ℓ^1 or p_ℓ^2 .*

Proof. Let us denote the total revenue by $\Pi = \Pi_{big} + \sum_{\ell=1}^{2L} \Pi_\ell$, where Π_{big} is the revenue obtained from the big customer and Π_ℓ is the revenue obtained from selling the items in gadget $\ell \in \{1, \dots, 2L\}$.

Consider a gadget $1 \leq \ell < 2L$. If $p_{\ell,2} + p_{\ell,3} > 2M + \frac{1}{2}a'_\ell$ then at most one of the customers requesting bundle $\{i_{\ell,2}\}$ or $\{i_{\ell,3}\}$ is a winner. In this case, we can bound the revenue on gadget ℓ from above by

$$\Pi_\ell \leq 2M - \frac{1}{2}a'_\ell + 2M - \frac{1}{2}a'_\ell + 8L \left(2M + \left(\frac{1}{2} + \frac{1}{8L} \right) a'_\ell \right) + M = 5M + (16M + 4a'_\ell)L.$$

On the other hand, if the customer requesting bundle $\{i_{\ell,2}\}$ and the customer requesting bundle $\{i_{\ell,3}\}$ are both winners, then each of the customers requesting $\{i_{\ell,2}, i_{\ell,3}\}$ cannot contribute more than $2M + \frac{1}{2}a'_\ell$ to the revenue. Therefore, also in this case the revenue due to the items in this gadget can be bounded by $\Pi_\ell \leq 5M + (16M + 4a'_\ell)L$.

With similar arguments, we can bound the revenue on gadget $2L$ by $\Pi_{2L} \leq 10M + (16M + 4a'_\ell)L$, and the big customer contributes at most $10ML + \frac{1}{8}a_{2L} + \frac{1}{8L}A$. Hence, the total revenue

$$\begin{aligned} \Pi &= \Pi_{big} + \sum_{\ell=1}^{2L} \Pi_\ell \leq 10ML + \frac{1}{8}a_{2L} + \frac{1}{8L}A + \sum_{\ell=1}^{2L} (5M + (16M + 4a'_\ell)L) + 5M \\ &\leq (20 + 32L)ML + 5M + 4L \sum_{\ell=1}^{2L} a_\ell + (8L^2 + \frac{1}{8})a_{2L} + \frac{1}{8L}A. \end{aligned}$$

Therefore, to have a revenue of at least $(20 + 32L)ML + 5M + 4L \sum_{\ell=1}^{2L} a_\ell + (8L^2 + \frac{1}{8})a_{2L} + \frac{1}{8L}A$, the revenues on each of the gadgets and the big bundle need to be equal to the before mentioned upper bounds. It is easy to verify that only the price vectors p_ℓ^1 and p_ℓ^2 satisfy this constraint. \square

We show that these price vectors p_ℓ^1 and p_ℓ^2 always lead to a pricing strategy that satisfies the monotonicity constraint.

Lemma 2.3.4. *Any pricing strategy that only uses price vectors p_ℓ^1 and p_ℓ^2 for each gadget in instance \mathcal{H} of the highway problem satisfies the monotonicity constraint.*

Proof. We know that $0 \leq a_1 \leq a_2 \leq \dots \leq a_{2L}$. Notice that both p_ℓ^1 and p_ℓ^2 consist of four item prices, and the sum of the item prices in p_ℓ^1 is $5M$ and in p_ℓ^2 it is $5M + \frac{1}{8L}a'_\ell$. Now consider an arbitrary bundle of size K , where $K \leq n = 8L$. Since only prices p_ℓ^1 and p_ℓ^2 are used for the gadgets, depending on the value of K modulo 4, we can easily determine lower and upper bounds on the price of any bundle of size K . To find the lower (upper) bound for the bundle price, it is sufficient to consider only prices in p_ℓ^1 (p_ℓ^2), because $p_{\ell,i}^1 \leq p_{\ell,i}^2$ for every $i \in \{1, \dots, 4\}$. Consider the following extensive case study, in which we determine the bounds for different values of K modulo 4.

$K \equiv 0 \pmod{4}$: If the bundle starts at item $i_{\ell,1}$ in some gadget ℓ then the total bundle price is at least $5M \frac{K}{4}$ and at most $(5M + \frac{1}{8L}a'_{2L}) \frac{K}{4}$. If the bundle starts at item $i_{\ell,2}$ in some gadget ℓ then the total bundle price is at least

$5M\frac{K}{4} + \frac{1}{2}a'_1 - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K}{4} + \frac{1}{2}a'_{2L} - \frac{1}{2}a'_1$. If the bundle starts at item $i_{\ell,3}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K}{4} + \frac{1}{2}a'_1 - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K}{4} + \frac{1}{2}a'_{2L} - \frac{1}{2}a'_1$. If the bundle starts at item $i_{\ell,4}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K}{4}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K}{4}$. Concluding, for a bundle of size $K \equiv 0 \pmod{4}$, the bundle price is at least $5M\frac{K}{4} + \frac{1}{2}a'_1 - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K}{4} + \frac{1}{2}a'_{2L} - \frac{1}{2}a'_1$.

$K \equiv 1 \pmod{4}$: If the bundle starts at item $i_{\ell,1}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-1}{4} + 2M - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-1}{4} + 2M - \frac{1}{2}a'_1$. If the bundle starts at item $i_{\ell,2}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-1}{4} + a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-1}{4} + (1 + \frac{1}{8L})a'_{2L}$. If the bundle starts at item $i_{\ell,3}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-1}{4} + 2M - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-1}{4} + 2M - \frac{1}{2}a'_1$. If the bundle starts at item $i_{\ell,4}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-1}{4} + M$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-1}{4} + M$. Concluding, for a bundle of size $K \equiv 1 \pmod{4}$, the bundle price is at least $5M\frac{K-1}{4} + a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-1}{4} + 2M - \frac{1}{2}a'_1$ (as M is large enough).

$K \equiv 2 \pmod{4}$: If the bundle starts at item $i_{\ell,1}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-2}{4} + 2M + \frac{1}{2}a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-2}{4} + 2M + (\frac{1}{2} + \frac{1}{8L})a'_{2L}$. If the bundle starts at item $i_{\ell,2}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-2}{4} + 2M + \frac{1}{2}a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-2}{4} + 2M + (\frac{1}{2} + \frac{1}{8L})a'_{2L}$. If the bundle starts at item $i_{\ell,3}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-2}{4} + 3M - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-2}{4} + 3M - \frac{1}{2}a'_1$. If the bundle starts at item $i_{\ell,4}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-2}{4} + 3M - \frac{1}{2}a'_{2L}$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-2}{4} + 3M - \frac{1}{2}a'_1$. Concluding, for a bundle of size $K \equiv 2 \pmod{4}$, the bundle price is at least $5M\frac{K-2}{4} + 2M + \frac{1}{2}a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-2}{4} + 3M - \frac{1}{2}a'_1$.

$K \equiv 3 \pmod{4}$: If The bundle starts at item $i_{\ell,1}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-3}{4} + 4M$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-3}{4} + 4M + \frac{1}{8L}a'_{2L}$. If The bundle starts at item $i_{\ell,2}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-3}{4} + 3M + \frac{1}{2}a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-3}{4} + 3M + (\frac{1}{2} + \frac{1}{8L})a'_{2L}$. If The bundle starts at item $i_{\ell,3}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-3}{4} + 5M - a'_{2L}$ and at most $(5M +$

$\frac{1}{8L}a'_{2L})\frac{K-3}{4} + 5M - a'_1$. If The bundle starts at item $i_{\ell,4}$ in some gadget ℓ then the total bundle price is at least $5M\frac{K-3}{4} + 3M + \frac{1}{2}a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-3}{4} + 3M + (\frac{1}{2} + \frac{1}{8L})a'_{2L}$. Concluding, for a bundle of size $K \equiv 3 \pmod{4}$, the bundle price is at least $5M\frac{K-3}{4} + 3M + \frac{1}{2}a'_1$ and at most $(5M + \frac{1}{8L}a'_{2L})\frac{K-3}{4} + 5M - a'_1$.

We summarize the resulting lower and upper bounds in Table 2.1. We use the lower and upper bound on the prices for a bundle of size K to determine whether the monotonicity constraint holds. To do this, the difference between the lower bound of a bundle of size $K + 1$ and the upper bound on a bundle of size K , that is, $LB(K + 1) - UB(K)$, should be positive for any bundle of size $K \leq 8L$.

Bundle size	LB	UB
$K \equiv 0 \pmod{4}$	$5M\frac{K}{4} + \frac{1}{2}a'_1 - \frac{1}{2}a'_{2L}$	$5M\frac{K}{4} - \frac{1}{2}a'_1 + (\frac{1}{2} + \frac{K}{4} \cdot \frac{1}{8L})a'_{2L}$
$K \equiv 1 \pmod{4}$	$5M\frac{K-1}{4} + a'_1$	$5M\frac{K-1}{4} + 2M - \frac{1}{2}a'_1 + \frac{K-1}{4} \cdot \frac{1}{8L}a'_{2L}$
$K \equiv 2 \pmod{4}$	$5M\frac{K-2}{4} + 2M + \frac{1}{2}a'_1$	$5M\frac{K-2}{4} + 3M - \frac{1}{2}a'_1 + \frac{K-2}{4} \cdot \frac{1}{8L}a'_{2L}$
$K \equiv 3 \pmod{4}$	$5M\frac{K-3}{4} + 3M + \frac{1}{2}a'_1$	$5M\frac{K-3}{4} + 5M - a'_1 + \frac{K-3}{4} \cdot \frac{1}{8L}a'_{2L}$

Table 2.1: Smallest and largest bundle prices for bundles with size K .

$K \equiv 0 \pmod{4}$: $LB(K + 1) - UB(K) =$

$$\begin{aligned}
 & 5M\frac{(K+1)-1}{4} + a'_1 - \left(5M\frac{K}{4} - \frac{1}{2}a'_1 + \left(\frac{1}{2} + \frac{K}{4} \frac{1}{8L}\right)a'_{2L}\right) \\
 &= \frac{3}{2}a'_1 - \left(\frac{1}{2} + \frac{K}{4} \frac{1}{8L}\right)a'_{2L} \geq \frac{3}{2}a'_1 - \left(\frac{1}{2} + \frac{8L}{4} \frac{1}{8L}\right)a'_{2L} \\
 &= \frac{3}{2}(a_1 + a_{2L}) - \frac{3}{4}(a_{2L} + a_{2L}) = \frac{3}{2}a_1 + \left(\frac{3}{2} - \frac{6}{4}\right)a_{2L} = \frac{3}{2}a_1 \geq 0.
 \end{aligned}$$

$K \equiv 1 \pmod{4}$: $LB(K + 1) - UB(K) =$

$$\begin{aligned}
 & 5M\frac{(K+1)-2}{4} + 2M + \frac{1}{2}a'_1 - \left(5M\frac{K-1}{4} + 2M - \frac{1}{2}a'_1 + \frac{K-1}{4} \cdot \frac{1}{8L}a'_{2L}\right) \\
 &= a'_1 - \frac{K-1}{4} \frac{1}{8L}a'_{2L} > a'_1 - \frac{8L}{4} \frac{1}{8L}a'_{2L} \\
 &= (a_1 + a_{2L}) - \frac{1}{4}(a_{2L} + a_{2L}) = a_1 + \frac{1}{2}a_{2L} \geq 0.
 \end{aligned}$$

$K \equiv 2 \pmod{4}$: $LB(K+1) - UB(K) =$

$$\begin{aligned}
 & 5M \frac{(K+1)-3}{4} + 3M + \frac{1}{2}a'_1 - \left(5M \frac{K-2}{4} + 3M - \frac{1}{2}a'_1 + \frac{K-2}{4} \cdot \frac{1}{8L}a'_{2L} \right) \\
 = & a'_1 - \frac{K-2}{4} \frac{1}{8L}a'_{2L} > a'_1 - \frac{8L}{4} \frac{1}{8L}a'_{2L} \\
 = & (a_1 + a_{2L}) - \frac{1}{4}(a_{2L} + a_{2L}) = a_1 + \frac{1}{2}a_{2L} \geq 0.
 \end{aligned}$$

$K \equiv 3 \pmod{4}$: $LB(K+1) - UB(K) =$

$$\begin{aligned}
 & 5M \frac{(K+1)}{4} + \frac{1}{2}a'_1 - \frac{1}{2}a'_{2L} - \left(5M \frac{K-3}{4} + 5M - a'_1 + \frac{K-3}{4} \cdot \frac{1}{8L}a'_{2L} \right) \\
 = & \frac{3}{2}a'_1 - \left(\frac{1}{2} + \frac{K-3}{4} \cdot \frac{1}{8L} \right) a'_{2L} > \frac{3}{2}a'_1 - \left(\frac{1}{2} + \frac{8L}{4} \cdot \frac{1}{8L} \right) a'_{2L} \\
 = & \frac{3}{2}(a_1 + a_{2L}) - \frac{3}{4}(a_{2L} + a_{2L}) = \frac{3}{2}a_1 + \left(\frac{3}{2} - \frac{6}{4} \right) a_{2L} = \frac{3}{2}a_1 \geq 0.
 \end{aligned}$$

We verified that this difference is positive in all four cases, hence the monotonicity constraint is satisfied. \square

Theorem 2.3.5. *The highway problem with monotonicity constraint is NP-hard.*

Proof. For the first part of the proof assume there is a set $S \subseteq \{1, \dots, 2L\}$ such that $\sum_{\ell \in S} a_\ell = A$ and $|S| = L$. Let the price vector of gadget ℓ be p_ℓ^1 if $\ell \notin S$, and p_ℓ^2 if $\ell \in S$. The revenue of every gadget ℓ is $(5 + 16L)M + 4L(a_\ell + a_{2L})$, independent of which price vector is used, and there is an additional revenue in gadget $2L$ of $5M$. Given the pricing strategy for all items, the big customer contributes

$$\begin{aligned}
 & \sum_{\ell \in S} \left(5M + \frac{1}{8L}(a_\ell + a_{2L}) \right) + \sum_{\ell \notin S} (5M) = \sum_{\ell=1}^{2L} 5M + \sum_{\ell \in S} \left(\frac{1}{8L}(a_\ell + a_{2L}) \right) \\
 = & 10LM + \frac{1}{8L}A + \frac{1}{8}a_{2L}.
 \end{aligned}$$

This yields a total revenue of

$$\begin{aligned}
 & \sum_{\ell=1}^{2L} (5M + 16LM + 4L(a_\ell + a_{2L})) + 5M + 10LM + \frac{1}{8L}A + \frac{1}{8}a_{2L} \\
 = & (20 + 32L)ML + 5M + 4L \sum_{\ell=1}^{2L} a_\ell + \left(8L^2 + \frac{1}{8} \right) a_{2L} + \frac{1}{8L}A.
 \end{aligned}$$

For the converse, assume there is a feasible solution to the highway pricing problem with a total revenue of at least $(20+32L)ML+5M+4L\sum_{\ell=1}^{2L}a_{\ell}+(8L^2+\frac{1}{8})a_{2L}+\frac{1}{8L}A$. By Lemma 2.3.3, we know that we use either p_{ℓ}^1 or p_{ℓ}^2 for each gadget ℓ . Notice that to obtain the maximum revenue on gadget $2L$, we use pricing p_{2L}^1 . Let set $S = \{\ell \in \{1, \dots, 2L\} : p_{\ell} = p_{\ell}^2\}$. Then, the actual payment of the big customer can also be written as

$$\sum_{\ell \in S} \left(5M + \frac{1}{8L}(a_{\ell} + a_{2L}) \right) + \sum_{\ell \notin S} (5M) = 10ML + \frac{1}{8L}|S|a_{2L} + \frac{1}{8L} \sum_{\ell \in S} a_{\ell}.$$

From the arguments in the proof of Lemma 2.3.3, we know that this should be equal to her valuation $10ML + \frac{1}{8}a_{2L} + \frac{1}{8L}A$. That is, $\sum_{\ell \in S} a_{\ell} = (L - |S|)a_{2L} + A$. We claim that $|S| = L$. To prove this, suppose it is not true. First, assume that $|S| < L$, then $\sum_{\ell \in S} a_{\ell} = (L - |S|)a_{2L} + A > \sum_{\ell=1}^{2L-1} a_{\ell} + A$, which is not possible as $2L \notin S$, so $|S| \geq L$. Now, assume $|S| > L$, then $\sum_{\ell \in S} a_{\ell} = (L - |S|)a_{2L} + A < -a_{2L} + A < 0$, as $A < a_{2L}$. This is also not possible as all integers a_{ℓ} are nonnegative. Therefore, we can conclude that $|S| = L$ and consequently, $A = \sum_{\ell \in S} a_{\ell}$. \square

2.3.2 Approximation algorithm

We cannot apply the PTAS from Theorem 2.2.7 to the highway problem, as this crucially requires the monotonicity constraint for arbitrary subsets of items. Nevertheless, we derive an $O(\ln B)$ -approximation algorithm for the highway pricing problem with monotonicity constraint. We denote by a ρ -approximation algorithm an algorithm that produces a solution with value at least $1/\rho$ times the optimal solution value. Before we present the algorithm, we present approximation guarantees for two special cases. First, for the case in which all bundles contain at least two items. Second, we consider the case in which all bundles contain only one item, and there is a bound on the ratio of the valuations between any two customers.

Lemma 2.3.6. *The highway problem with monotonicity constraint in which all bundles have size at least two is approximable within a factor of 3 by optimal uniform pricing.*

Proof. Consider an optimal solution with revenue Π^* and let p_{\max}^* be the highest item price in this solution. We claim that pricing all items at $p_{\max}^*/3$, yields a revenue of at least $\Pi^*/3$. Clearly, an optimal uniform pricing is at least as good as the uniform $p_{\max}^*/3$ pricing.

We claim that any winner $j \in W$ in the optimal pricing remains a winner for the uniform pricing at level $p_{\max}^*/3$. Let $|I_j| = \ell$. Then the valuation for customer j is at least $b_j \geq \lfloor \ell/2 \rfloor p_{\max}^*$, as by the monotonicity constraint the total price of any two consecutive items in an optimal solution is at least p_{\max}^* and customer j can afford the corresponding bundle I_j . In the uniform $p_{\max}^*/3$ pricing, the total bundle price is $\ell p_{\max}^*/3$, which is not more than $\lfloor \ell/2 \rfloor p_{\max}^*$, for $\ell \geq 2$. In an optimal pricing, bundle I_j is priced at most at ℓp_{\max}^* , whereas in our uniform pricing, we get $\ell p_{\max}^*/3$. Hence, pricing all items to $p_{\max}^*/3$ yields a revenue of at least $\Pi^*/3$. The optimal uniform pricing yields at least the same revenue, which proves the claim. \square

The above lemma shows that whenever all bundles contain at least two items, we have a constant approximation. Now, we consider only instances in which bundles consist of exactly one item. Moreover, we restrict ourselves to instances in which $b_j/b_k \leq \beta$, for any two customers j and k . The value of $\beta \geq 1$ will be defined later.

Lemma 2.3.7. *The highway problem with monotonicity constraint, restricted to instances in which each bundle contains exactly one item and $b_j/b_{j'} \leq \beta$ for any two customers $j, j' \in J$, admits a linear time β -approximation algorithm.*

Proof. We price each item uniformly at $p = \min_{j \in J} \{b_j\}$. Since for any two customers j and j' it holds that $b_j/b_{j'} \leq \beta$ and every bundle contains one item, we lose at most a factor of β on the contribution of every customer. Hence, pricing all items at p yields a revenue of at least Π^*/β . \square

Theorem 2.3.8. *The best uniform pricing yields a solution with revenue at least $\Pi^*/(3 + e \ln B)$ for the highway problem with monotonicity constraint. The time needed to find this solution is $O(n^2 m)$.*

Proof. Consider an optimum solution satisfying the monotonicity constraint. Let Π_L^* denote the revenue of customers whose bundles are of size at least two. Furthermore, for any $r \in R = \{1, \dots, \lceil \log_\beta B \rceil + 1\}$, let Π_r^* denote the revenue of customers whose bundles are of size one with valuation $\beta^{r-1} \leq b_j < \beta^r$ in this solution. Then $\Pi^* = \Pi_L^* + \sum_{r \in R} \Pi_r^*$.

Let $\tilde{\Pi}_L$ denote the revenue obtained by the best uniform pricing and $\tilde{\Pi}_r$ denote the revenue obtained by the best uniform pricing strategy for the customers in $J_r = \{j \in J : |I_j| = 1 \text{ and } \beta^{r-1} \leq b_j < \beta^r\}$, for all $r \in R$. By Lemma 2.3.7, we have that $\tilde{\Pi}_r \geq \Pi_r^*/\beta$ and thus

$$\max_{r \in R} \{\tilde{\Pi}_r\} \geq \sum_{r \in R} \frac{\Pi_r^*}{\beta \log_\beta B}.$$

Moreover, from Lemma 2.3.6, it follows that $\tilde{\Pi}_L \geq \Pi_L^*/3$. Hence, the solution yields a revenue of

$$\max\{\tilde{\Pi}_L, \tilde{\Pi}_r : r \in R\} \geq \frac{\Pi^*}{3 + \beta \log_\beta B}.$$

The revenue from the approximation algorithm is maximized for the value of β for which the derivative of the approximation ratio $(3 + \beta \log_\beta B)^{-1}$, with respect to β , is equal to 0. This is true for $\beta = e$ and therefore we get

$$\max\{\tilde{\Pi}_L, \tilde{\Pi}_r : r \in R\} \geq \frac{\Pi^*}{3 + e \ln B}.$$

To see the claim on the time complexity, note that to find an optimal uniform pricing, we need to consider at most n different prices, namely $b_j/|I_j|$ for all $j \in J$. For each price, we need to compute the set of winning customers and the revenue obtained on this price, which can be done in $O(nm)$ time. So, the best uniform price can be computed in $O(n^2m)$ time. \square

2.4 Conclusion

The monotonicity constraint introduced in this chapter forces prices to be of the same magnitude for all items, thus it can very well be applied for companies that sell comparable items. With the introduction of this monotonicity constraint on the bundle prices, the single item pricing problem is strongly NP-hard. Also, we show that a polynomial time approximation scheme exists for this particular problem, breaking the semi-logarithmic inapproximability result by Demaine et al. (2006).

In contrast, the prices are not forced to be of the same magnitude in the highway problem, and as proven in this chapter, this problem remains NP-hard. Being unable to close the gap between (weak) NP-hardness on the one hand and only logarithmic polynomial-time approximation algorithms on the other hand, we search for a different approach in the next chapter. There, we use the average valuations of the customers to obtain an approximation algorithm, and introduce a monotonicity constraint on the valuations instead of on the prices as was done in this chapter.

Chapter 3

Inhomogeneity of Average Valuations

In this chapter¹ we consider the single item pricing problem with unlimited supply. Customers request arbitrary subsets of a given set of items that are sold by a company. This problem is in general known to be inapproximable by a semi-logarithmic factor in the number of customers n as shown by Demaine et al. (2006). We relate the tractability of the problem to structural properties of customers' valuations. Then, the inapproximability result is no longer valid as soon as the inhomogeneity is bounded by a constant. Moreover, we show that the problem becomes strongly NP-hard as soon as the average valuations of customers are not homogeneous, even under further restrictions such as monotonicity of valuations.

We address the highway problem afterwards. This problem asks for prices to be determined for segments of a single highway such as to maximize the revenue obtainable from a given set of customers with known valuations. The problem is (weakly) NP-hard and a recent quasi-polynomial time approximation scheme by Elbassioni, Sitters, and Zhang (2007) suggests that a PTAS might be in reach. We show that also this problem becomes NP-hard as soon as the average valuations of customers are not homogeneous, even under further restrictions such as monotonicity of valuations. Moreover, we apply the efficient approximation algorithm, parameterized along the inhomogeneity of customers' average valuations, that is derived for the single item pricing problem with arbitrary bundles.

¹This chapter is based on Grigoriev, van Loon, and Uetz (2008b).

3.1 Introduction

Let $I = \{1, \dots, m\}$ represent the set of items for sale, where each item is available in unlimited supply. Let $J = \{1, \dots, n\}$ denote the set of potential single-minded customers. Every customer $j \in J$ requests a subset of the items, the bundle, denoted $I_j \subseteq I$. The maximum amount customer j wants to pay for subset I_j , her valuation b_j , is publicly known. We assume $b_j > 0$, for otherwise that customer can be deleted from the instance. Given a vector of prices $p = (p_1, \dots, p_m)$, containing one price for each item, denote by $W = \{j \in J : \sum_{i \in I_j} p_i \leq b_j\}$ the set of *winners*.

Definition 3.1.1. *The single item pricing problem with arbitrary bundles asks for a vector of prices $p = (p_1, \dots, p_m) \in \mathbb{R}_+^m$, one for each item, such that the total revenue Π extracted from the set W of winners is maximal, where $\Pi = \sum_{j \in W} \sum_{i \in I_j} p_i$.*

In this chapter we determine the maximum deviation between customers' average valuations, referred to as the inhomogeneity of average valuations.

Definition 3.1.2 (Inhomogeneity of average valuations). *For any instance of the single item pricing problem, define $\bar{b}_j = b_j/|I_j|$ as the average (per item) valuation of customer j , and define the inhomogeneity of valuations as*

$$\alpha = \max_{j, j' \in J} \left\{ \frac{\bar{b}_j}{\bar{b}_{j'}} \right\}.$$

Notice that $\alpha \geq 1$, and that the problem becomes trivial as soon as the valuations are *homogeneous*, that is, $\alpha = 1$, since this corresponds to the case where all customers' valuations per segment are identical; see also Section 3.2.1.

Our first result shows that, in contrast to the trivially solvable homogeneous case, the problem with inhomogeneity of valuations is strongly NP-hard. While this does not sound very surprising, the main point is that this NP-hardness result holds even if the inhomogeneity α is bounded from above by any constant $1 + \varepsilon$. In some sense, we thereby delineate the borderline between triviality and NP-hardness for the single item pricing problem.

Furthermore, the NP-hardness result remains true even if we impose further restrictions on customers' valuations, such as *monotonicity*, that is,

$$b_j \leq b_{j'} \quad \text{for all } I_j \subseteq I_{j'},$$

and *monotonicity of average valuations*, that is,

$$\bar{b}_j = \frac{b_j}{|I_j|} \geq \frac{b_{j'}}{|I_{j'}|} = \bar{b}_{j'} \quad \text{for all } I_j \subseteq I_{j'}.$$

Our second result is a parametric approximation algorithm for the single item pricing problem that complements the NP-hardness result. The proposed algorithm has performance guarantee $O(\ln \alpha)$ and computation time $O(n(\log n + m))$, where the constant hidden in the O -notation of the performance bound is not more than e . More specifically, it is easy to see that an α -approximation exists, and for large values of α we show how to improve this bound to $1 + \ln \alpha + \varepsilon$ for any $\varepsilon > 0$. In this context, notice that if there exists any constant upper bound on the inhomogeneity α then the semi-logarithmic inapproximability result of Demaine et al. (2006) for that problem is not longer valid. We believe that such a constant bound is not unreasonable in practical applications, especially if the company sells comparable items. Also, note that $\alpha \leq m$ for the case of monotone and decreasing average valuations.

Finally, we briefly comment on the fact that the $O(\ln \alpha)$ -approximation result also holds for the *highway problem*. Here, the items represent consecutive segments of a single highway, which can be regarded as edges on a simple path. Customers are interested in subpaths of this path. Notice that as soon as the inhomogeneity α of customers' valuations is bounded by some constant, we have a constant-factor approximation algorithm. Similarly as for the problem with arbitrary bundles, we delineate the border between triviality if $\alpha = 1$ and (weak) NP-hardness if the inhomogeneity α is bounded from above by $1 + \varepsilon$.

3.2 Arbitrary bundles

In this section, we discuss the complexity of the single item pricing problem with arbitrary bundles. Furthermore, we propose a $O(\ln \alpha)$ -approximation algorithm.

3.2.1 Complexity

We start with the short argument that the single item pricing problem with homogeneous average valuations is trivially solvable: consider the average

valuation \bar{b} , which is, by homogeneity, the same for each customer, and define the price $p_i = \bar{b}$ for every item $i \in I$. Clearly, each customer contributes her entire valuation to the revenue, and the obtained solution is optimal.

Surprisingly enough, even if we allow only arbitrarily small deviations from homogeneous average valuations, the single item pricing problem becomes NP-hard. More specifically, in Theorem 3.2.2 we show that the problem with inhomogeneous average valuations is strongly NP-hard under the following three conditions.

Condition 3.1. *The inhomogeneity $\alpha \leq 1 + \varepsilon$, where $\varepsilon > 0$ is an arbitrary constant.*

Condition 3.2. *The valuations of the customers are monotone, that is, $b_j \leq b_{j'}$ for any $j, j' \in J$ such that $I_j \subseteq I_{j'}$.*

Condition 3.3. *The average valuations of the customers are monotone decreasing, that is, $\bar{b}_{j'} \leq \bar{b}_j$ for any $j, j' \in J$ such that $I_j \subseteq I_{j'}$.*

To prove this, we use a reduction from the NP-hard problem INDEPENDENTSET (Garey and Johnson 1979). Given a graph $G = (V, E)$ and integer $s \leq |V|$, does there exist a set of vertices that are pairwise non-adjacent with cardinality at least s ? We define an instance \mathcal{I} of the pricing problem as follows. Given an $\varepsilon > 0$, let $M \geq \max\{1/\varepsilon, s + 1/2\}$. For every vertex $v \in V$ we create two *vertex-items*, v^1 and v^2 , and for every edge $e \in E$ we introduce two *edge-items*, e^1 and e^2 . Thus, $I = V \cup E$. Every vertex- and edge-item is requested by $2M^2 + 2M - 1$ customers with valuation $M + 1$. For every vertex $v \in V$, there is one customer interested in bundle $\{v^1, v^2\}$ and similarly, for every edge $e \in E$, there is one customer interested in bundle $\{e^1, e^2\}$. These customers have valuation $2M + 2 - 1/M$. There is one customer interested in item x with valuation $M + 1$, and there are two customers interested in bundle $\{y\}$ of size M with valuation M^2 . Also, there are two customers requesting bundle $\{x, y\}$ (of size $M + 1$) with valuation $M^2 + M$. Then, for every edge $e = \{u, v\} \in E$, there is one customer interested in bundle $\{u^1, u^2, v^1, v^2, e^1, e^2\} \cup \{x\}$ with valuation $7M + 6 - 2/M$. And finally, one customer requests all vertex-items and item x , that is, bundle $\{v^1, v^2 : v \in V\} \cup \{x\}$, with valuation $(2M + 2 - 1/M)|V| + M + (1/M)s$. An example of an instance of the single item pricing problem with arbitrary bundles is displayed in Figure 3.1, where graph G has vertices $\{u, v, w\}$ and edges $e = \{u, v\}$ and $f = \{v, w\}$.

Let us give a short intuition as to why we need these particular bundles. The bundles on the vertex- and edge-items determine which vertices are in

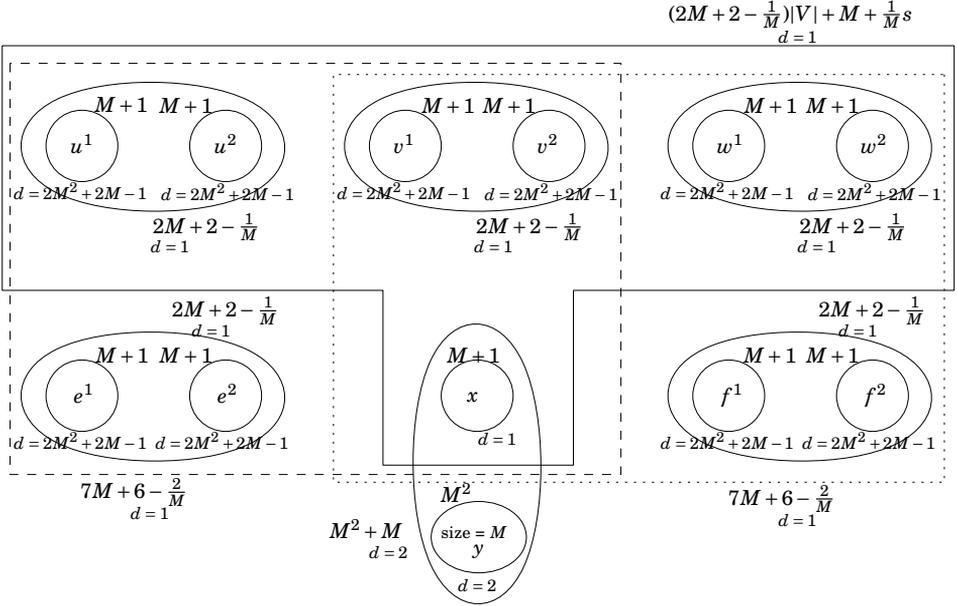


Figure 3.1: Example of instance \mathcal{I} of the single item pricing problem with arbitrary bundles reduced from graph $G = (\{u, v, w\}, \{e = \{u, v\}, f = \{v, w\}\})$.

the independent set of G and bundle $\{u^1, u^2, v^1, v^2, e^1, e^2\} \cup \{x\}$ assures later that a feasible solution to the single item pricing problem corresponds to an independent set in G . Bundle $\{v^1, v^2 : v \in V\} \cup \{x\}$ assures that a feasible solution to the pricing problem corresponds to an independent set of cardinality s . Finally, bundles $\{x\}$, $\{y\}$ and $\{x, y\}$ are present to fulfill the three conditions required in this theorem.

Lemma 3.2.1. *Instance \mathcal{I} satisfies Conditions 3.1, 3.2 and 3.3.*

Proof. The single-item bundles have the largest average valuation of $M + 1$, and bundles $\{y\}$ and $\{x, y\}$ have the smallest average valuation of M , thus $\alpha = 1 + 1/M \leq 1 + \epsilon$, as $M \geq 1/\epsilon$. It is fairly easy to see that Condition 3.2 holds as the valuation of a customer for some bundle is strictly smaller than the valuation of a customer requesting a superset of this particular bundle. The verification of Condition 3.3 is a little more troublesome. First note that for bundle $\{i^1\}$ or $\{i^2\}$ the condition holds regarding bundle $\{i^1, i^2\}$ as $M + 1 > M + 1 - \frac{1}{2M}$, for all $i \in I$. Also, for bundles $\{i^1, i^2\}$ and $\{u^1, u^2, v^1, v^2, e^1, e^2\} \cup \{x\}$

we have

$$\frac{2M+2-\frac{1}{M}}{2} = M+1 - \frac{1}{2M} \geq M + \frac{6}{7} - \frac{2}{7M} = \frac{7M+6-\frac{2}{M}}{7},$$

where the inequality holds as $M \geq s + 1/2 \geq 3/2$. Furthermore, for bundles $\{i^1, i^2\}$ and $\{v^1, v^2 : v \in V\} \cup \{x\}$ we have

$$\frac{2M+2-\frac{1}{M}}{2} = M+1 - \frac{1}{2M} \geq \frac{(2M+2-\frac{1}{M})|V| + M + \frac{1}{M}s}{2|V|+1},$$

as $M \geq s + 1/2$. The average valuation of bundle $\{x\}$ is $M + 1$, thus as a consequence of the above calculation, the condition holds for bundle $\{x\}$ in either $\{v^1, v^2 : v \in V\} \cup \{x\}$ or $\{u^1, u^2, v^1, v^2, e^1, e^2\} \cup \{x\}$. Finally, the average valuation of $\{x\}$ is $M + 1$ which is clearly greater than the average valuation M for $\{x, y\}$, and as this is equal to the average valuation for $\{y\}$ we have verified that Condition 3.3 holds. \square

In the following proof, we define Π_i as the revenue obtained from the customers requesting a bundle from set $\{i^1, i^2, \{i^1, i^2\}\}$ for all $i \in I = V \cup E$. We define Π_e as the revenue from the customers requesting $\{u^1, u^2, v^1, v^2, e^1, e^2\} \cup \{x\}$ for some $e = \{u, v\} \in E$. We define Π_{xy} as the revenue received from customers requesting a bundle from set $\{\{x\}, \{y\}, \{x, y\}\}$, and finally, Π_V represents the revenue obtained from the customers requesting $\{v^1, v^2 : v \in V\} \cup \{x\}$. Obviously, the total revenue is $\Pi = \sum_{i \in I} \Pi_i + \sum_{e \in E} \Pi_e + \Pi_{xy} + \Pi_V$. Let C be a constant equal to $(|V| + |E|)(4M^3 + 8M^2 + 2M - 2) + |E|(7M + 6 - 2/M) + |V|(2M + 2 - 1/M) + 4M^2 + 4M$.

Theorem 3.2.2. *The single item pricing problem with arbitrary bundles is strongly NP-hard, even when restricted to the instances satisfying Conditions 3.1, 3.2 and 3.3.*

Proof. We claim that there exists an independent set $V' \subseteq V$ of size $|V'| = s$ if and only if there exists a feasible solution to instance \mathcal{S} of the single item pricing problem with arbitrary bundles with revenue at least $C + (1/M)s$.

Given an independent set $V' \subseteq V$ of size $|V'| = s$. Define $E_0 = \{e = \{u, v\} \in E : u, v \notin V'\}$. Let $p_i = (p_{i^1}, p_{i^2})$ be defined by $p_i = (M + 1, M + 1)$ if $i \in V \cap V'$ or $i \in E \cap E_0$ and $p_i = (M + 1 - \frac{1}{2M}, M + 1 - \frac{1}{2M})$ if $i \in V \setminus V'$ or $i \in E \setminus E_0$. Also, let $p_x = M$ and $p_y = M^2$, where p_y denotes the sum of all M item prices in bundle $\{y\}$. Under this pricing strategy, we see that $\Pi_i = 4M^3 + 8M^2 + 2M - 2$ for all $i \in I = V \cup E$, irrespective of which pricing is used for item i , and

$\Pi_{x,y} = M + 2M^2 + 2(M^2 + M) = 4M^2 + 3M$. Every edge $e = \{u, v\} \in E$ contains one item priced at $(M + 1, M + 1)$ and two at $(M + 1 - \frac{1}{2M}, M + 1 - \frac{1}{2M})$ by definition of the pricing and set E_0 . As $p_x = M$, we have

$$\Pi_e = 2(M + 1) + 4\left(M + 1 - \frac{1}{2M}\right) + M = 7M + 6 - \frac{2}{M}.$$

The customer requesting all vertex-items and item x spends

$$\left(2M + 2 - \frac{1}{M}\right)|V \setminus V'| + (2M + 2)|V'| + p_x = \left(2M + 2 - \frac{1}{M}\right)|V| + M + \frac{1}{M}s.$$

Then, the total revenue is

$$\Pi = (|V| + |E|)\Pi_i + |E|\Pi_e + \Pi_{x,y} + \Pi_V = C + \frac{1}{M}s.$$

For the converse, we are given a solution to instance \mathcal{S} with revenue at least $C + (1/M)s$. First, we consider $\Pi_{x,y}$. If the customer requesting bundle $\{x, y\}$ is not a winner, the maximum revenue is $M + 1 + 2M^2$. Otherwise, let p_x be the price for item x . Then, the maximum revenue is $p_x + 2(M^2 + M - p_x) + 2(M^2 + M)$, where $p_x \in [M, M + 1]$ such that all customers are winners. Then, $\Pi_{x,y} \leq 4M^2 + 3M$, which is obtained if $p_x = M$. For every item $i \in I$, we have

$$\Pi_i = \max\{2(2M^2 + 2M - 1)(M + 1), (2M^2 + 2M - 1 + 1)(2M + 2 - 1/M)\}.$$

This is equal to $\Pi_i = 4M^3 + 8M^2 + 2M - 2$. Clearly, for every $e \in E$, the revenue Π_e is at most the valuation $7M + 6 - 2/M$. Now, we know that the revenue from the customer requesting bundle $\{v^1, v^2 : v \in V\} \cup \{x\}$ is

$$\Pi_V = \Pi - (|V| + |E|)\Pi_i - |E|\Pi_e - \Pi_{x,y} \geq (2M + 2 - 1/M)|V| + M + (1/M)s.$$

Thus, the minimum revenue is at least equal to the valuation. As this customer cannot contribute more than the valuation, it should be equality throughout. This also means that all other revenues described above attain their maximum, thus $p_x = M$ and $p_y = M^2$. Now, let $V' = \{v \in V : p_v = (M + 1, M + 1)\}$ and $E_0 = \{e \in E : p_e = (M + 1, M + 1)\}$. As $\Pi_e = 7M + 6 - 2/M$ and $p_x = M$ for all $e = \{u, v\} \in E$, we know that either $u \in V'$ and $v \notin V'$, $e \notin E_0$, or $v \in V'$ and $u \notin V'$, $e \notin E_0$, or $e \in E_0$ and $u, v \notin V'$. Thus, for each edge, either one vertex is in V' or both are not in. Hence, V' is an independent set. Furthermore, the customer requesting bundle $\{v^1, v^2 : v \in V\} \cup \{x\}$ pays

$$(2M + 2 - 1/M)|V \setminus V'| + (2M + 2)|V'| + p_x = (2M + 2 - 1/M)|V| + M + (1/M)|V'|.$$

As this payment is equal to the revenue, which in turn has to be equal to the valuation, we know that $|V'| = s$. \square

3.2.2 Approximation algorithm

The idea for the approximation algorithm is as follows. We partition the set of customers J into $O(\ln \alpha)$ subsets S_1, \dots, S_K , such that in each subset any two customers have average valuations different from each other by at most a constant factor $\delta > 1$. Denote by Π_k^* the maximum revenue for the single item pricing problem restricted to the set of customers S_k , referred to as S_k -restricted problem. Then $\sum_{k=1}^K \Pi_k^*$ is clearly an upper bound for the optimum Π^* of the original problem. Therefore, the highest maximum revenue $\max_{k=1, \dots, K} \{\Pi_k^*\}$ over all restricted problems is at least Π^*/K . Next, from the fact that the inhomogeneity of the average valuations in S_k is bounded by at most a factor of δ , we derive that for the S_k -restricted problem there exist a price vector generating revenue of at least Π_k^*/δ . Thus, taking the pricing vector yielding the highest revenue over all restricted problems, we generate a total revenue of at least $\Pi^*/\delta K$. Finally, we optimize the performance guarantee over parameters K and δ , in Lemma 3.2.3 and Theorem 3.2.5, respectively.

Consider Algorithm 3.1 in which we create sets S_1, \dots, S_K and determine a price vector p yielding revenue $\tilde{\Pi}$.

Lemma 3.2.3. *For any $\delta > 1$, the number of subsets K is at most $1 + \ln \alpha / \ln \delta$.*

Proof. By definition of the inhomogeneity α , we have $\bar{b}_j \leq \alpha \bar{b}_{j'}$ for every pair of customers $j, j' \in J$. We show by induction that for any $k = 1, \dots, K$ the ratio between the highest and lowest average valuation of customers in J_k , \bar{b}_k^{\max} and \bar{b}_k^{\min} respectively, is at most α / δ^{k-1} . This holds trivially for $k = 1$. Now, suppose that the claim is true up to $k - 1$ for some $k = 2, \dots, K$. Then, we have

$$\frac{\bar{b}_k^{\max}}{\bar{b}_k^{\min}} < \frac{\bar{b}_1^{\max}}{\delta^{k-1} \bar{b}_1^{\min}} \leq \frac{\alpha}{\delta^{k-1}},$$

where the first inequality holds because $\bar{b}_k^{\min} > \delta^{k-1} \bar{b}_1^{\min}$ by definition of S_k and $\bar{b}_k^{\max} = \bar{b}_1^{\max}$ by definition of J_k , and the second inequality holds by the induction hypothesis. Consequently, $K \leq 1 + \log_\delta \alpha = 1 + \ln \alpha / \ln \delta$. \square

Lemma 3.2.4. *In the S_k -restricted problem, price vector p^k yields a revenue $\tilde{\Pi}_k \geq \Pi_k^*/\delta$.*

Proof. For any $k = 1, \dots, K$, pricing p^k determined in Algorithm 3.1 assures that all customers in S_k are winners. By definition of set S_k , $\max_{j \in S_k} \bar{b}_j / \min_{j \in S_k} \bar{b}_j \leq \delta$, therefore the revenue of the solution is at least Π_k^*/δ . \square

Algorithm 3.1: Approximation algorithm

Initialize $J_1 = J$, $\tilde{\Pi} = 0$, $W = J$;
Order the customer with respect to their average valuation in an increasing fashion;
 $\bar{b}_{\min} = \min_{j \in J} \{\bar{b}_j\}$;
foreach $k = 1, \dots, K$ **do**
 $S_k = \{j \in J_k : \bar{b}_j \leq \delta^k \bar{b}_{\min}\}$;
 $J_{k+1} = J_k \setminus S_k$;
 $\Pi_k = 0$;
 foreach $i = 1, \dots, m$ **do**
 $S_{ik} = \{j \in S_k : i \in I_j\}$;
 if $S_{ik} \neq \emptyset$ **then**
 $p_i^k = \min_{j \in S_{ik}} \{\bar{b}_j\}$;
 else
 p_i^k is chosen arbitrarily;
 end
 $\Pi_k += |S_{ik}| p_i^k$;
 end
 if $\Pi_k > \tilde{\Pi}$ **then**
 $\tilde{\Pi} = \Pi_k$, $p = p^k$ and $W = S_k$;
 end
end

Combining Lemma 3.2.3 and Lemma 3.2.4 and optimization of δ yields the following result.

Theorem 3.2.5. *Price vector p yields a total revenue of at least $\Pi^*/(e \ln \alpha + e)$ for the single item pricing problem, and it can be computed in $O((n+m) \log n)$ time.*

Proof. The total revenue generated by the best price vector p from $\{p^k : k = 1, \dots, K\}$ is

$$\tilde{\Pi} = \max_{k=1, \dots, K} \{\tilde{\Pi}_k\} \geq \max_{k=1, \dots, K} \left\{ \frac{\Pi_k^*}{\delta} \right\} \geq \frac{\Pi^*}{\delta K} \geq \frac{\Pi^*}{\delta \left(1 + \frac{\ln \alpha}{\ln \delta}\right)}.$$

This is maximized if

$$\ln \delta = \frac{-\ln \alpha + \sqrt{(\ln \alpha)^2 + 4 \ln \alpha}}{2} = \frac{-\ln \alpha + 2 \ln \alpha \sqrt{\frac{1}{4} + \frac{1}{\ln \alpha}}}{2} = \left(\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{\ln \alpha}} \right)^{-1}$$

thus for $\delta = \exp \left(\left(\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{\ln \alpha}} \right)^{-1} \right)$. Now, as $e \ln \alpha + e \geq \delta(1 + \frac{\ln \alpha}{\ln \delta})$ for all $\alpha > 1$, the total revenue is at least $\Pi^*/(e \ln \alpha + e)$.

We arrive at the computation time as follows. First, we order the customers according to their average valuation increasingly, which can be done in $O(n \log n)$ time. Then, for all $k = 1, \dots, K$, we use binary search to create set S_k in $O(\log n)$ time, and for all items $i = 1, \dots, m$ we determine the set of customers that request the item in $O(n)$ time, and the item price and the revenue in constant time. So, the total runtime is $O(n \log n + K(\log n + nm))$, which is in $O(n(\log n + m))$, as K depends on α . \square

There are several directions for improvement of the obtained approximate solution to the single item pricing problem. First, instead of the constructed price vectors $p^k, k = 1, \dots, K$, we can use price vectors maximizing the revenue in the S_k -restricted problems, with given set of winners $W = S_k$. Notice that, for any set of winners $W \subseteq J$, the price vector maximizing the revenue obtained from W can be found in polynomial time by solving a simple linear program; see Lemma 4.2.2. Unfortunately, this approach does not necessarily lead to any provable improvement of the performance guarantee.

Second, we can improve the performance guarantee when using a more careful analysis of the revenue generated by price vector p when applied to the entire set J instead of S_k only. By construction of the partition of J , for any two subsets S_k and $S_{k'}, k \leq k'$, the average valuation of any customer in S_k is at most the average valuation of a customer from $S_{k'}$. Therefore, for any $k = 1, \dots, K$, and for all $k' \geq k$, if $S_k \subseteq W$ then $S_{k'} \subseteq W$ as well. By definition of the subsets, the maximum average valuation in set S_{k+1} is at most δ times the maximum average valuation in set S_k . By combining this insight with Lemma 3.2.4, we have that the revenue generated by price vector p^k applied to the set of customers J is at least

$$\Pi_k = \frac{1}{\delta} \Pi_k^* + \frac{1}{\delta^2} \Pi_{k+1}^* + \dots + \frac{1}{\delta^{K-k+1}} \Pi_K^*, \forall k = 1, \dots, K.$$

These equalities can be equivalently represented by the following recurrent equation

$$\Pi_k = \frac{1}{\delta} \Pi_k^* + \frac{1}{\delta} \Pi_{k+1}, \forall k = 1, \dots, K-1, \quad (3.1)$$

with an additional equality

$$\bar{\Pi}_K = \frac{1}{\delta} \Pi_K^*. \quad (3.2)$$

Summing up all Equations (3.1) and (3.2) and dividing both sides by K , we derive

$$\bar{\Pi} = \frac{1}{K} \sum_{k=1}^K \Pi_k = \frac{1}{K\delta} \sum_{k=1}^K \Pi_k^* + \frac{1}{K\delta} \sum_{k=1}^K \Pi_k - \frac{1}{K\delta} \Pi_1.$$

Let $\Pi_1 = \phi \bar{\Pi}$. Since $\sum_{k=1}^K \Pi_k^* \geq \Pi^*$, we derive

$$\bar{\Pi} \geq \frac{1}{K\delta} \Pi^* + \frac{1}{\delta} \bar{\Pi} - \frac{1}{K\delta} \phi \bar{\Pi} \Leftrightarrow \bar{\Pi} \geq \frac{\Pi^*}{K(\delta-1) + \phi}.$$

Taking the maximum revenue over all price vectors $p^k, k = 1, \dots, K$, we obtain

$$\tilde{\Pi} = \max_{k=1, \dots, K} \{\Pi_k\} \geq \max\{\Pi_1, \bar{\Pi}\} \geq \max\left\{ \frac{\phi \Pi^*}{K(\delta-1) + \phi}, \frac{\Pi^*}{K(\delta-1) + \phi} \right\}.$$

This latter maximum is minimized for $\phi = 1$, yielding

$$\tilde{\Pi} = \max_{k=1, \dots, K} \{\Pi_k\} \geq \frac{\Pi^*}{\delta(1 + \frac{\ln \alpha}{\ln \delta}) - \frac{\ln \alpha}{\ln \delta}},$$

Clearly, the best price vector yields a total revenue of at least $\Pi^*/(\delta(1 + \frac{\ln \alpha}{\ln \delta}) - \frac{\ln \alpha}{\ln \delta})$. Now, we know that

$$\delta \left(1 + \frac{\ln \alpha}{\ln \delta} \right) - \frac{\ln \alpha}{\ln \delta} < \delta \ln \alpha + \delta.$$

To see this, consider function $f(\delta) = \ln \delta + 1/\delta$. As the derivative $f'(\delta) = \delta^{-1} - \delta^{-2} > 0$ for all $\delta > 1$, function $f(\delta)$ is monotone increasing, and consequently, $f(\delta) > f(1) = 1$ for all $\delta > 1$. Then,

$$\begin{aligned} \ln \delta + \frac{1}{\delta} > 1 &\Leftrightarrow \delta \ln \delta > \delta - 1 \Leftrightarrow \delta \ln \alpha > \delta \frac{\ln \alpha}{\ln \delta} - \frac{\ln \alpha}{\ln \delta} \\ &\Leftrightarrow \delta \ln \alpha + \delta > \delta \left(1 + \frac{\ln \alpha}{\ln \delta} \right) - \frac{\ln \alpha}{\ln \delta}. \end{aligned}$$

Given $\varepsilon > 0$, let $\delta = 1 + \varepsilon/(\ln \alpha + 1)$. Then,

$$\delta \ln \alpha + \delta = \left(1 + \frac{\varepsilon}{\ln \alpha + 1} \right) \ln \alpha + \left(1 + \frac{\varepsilon}{\ln \alpha + 1} \right) = 1 + \ln \alpha + \varepsilon,$$

and we arrive at the following theorem.

Theorem 3.2.6 (Improved Bound). *Price vector p yields a total revenue of at least $\Pi^*/(1 + \ln \alpha + \varepsilon)$ for the single item pricing problem for any $\varepsilon > 0$, and it can be computed in $O(n(\log n + m))$ time.*

3.3 Highway problem

We consider the *highway problem*, introduced by Guruswami et al. (2005). Let $I = \{1, \dots, m\}$ represent the highway segments, and regard them as consecutive edges on a simple path with unlimited capacity. Let $J = \{1, \dots, n\}$ denote the set of potential customers. Every customer $j \in J$ requests a sub-path of the highway, denoted $I_j \subseteq I$, and we assume that each I_j is of the form $I_j = \{k, k + 1, \dots, \ell\}$, where $1 \leq k \leq \ell \leq m$. The valuation b_j for traveling subpath I_j is publicly known. This is quite reasonable when assuming that the valuation is a monetary expression for the time saving that can be realized by using the highway instead of the next-fastest alternative route. We assume $b_j > 0$, for otherwise that customer can be deleted from the instance. Given a vector of prices $p = (p_1, \dots, p_m)$, containing one price for each highway segment, denote by $W = \{j \in J : \sum_{i \in I_j} p_i \leq b_j\}$ the set of *winners*.

Definition 3.3.1. *The highway problem asks for a vector of prices $p \in \mathbb{R}_+^m$, one for each segment of the highway, such that the total revenue Π extracted from the set W of winners is maximal, where $\Pi = \sum_{j \in W} \sum_{i \in I_j} p_i$.*

Intrigued by the gap between (weak) NP-hardness on the one hand, and only logarithmic polynomial-time approximation algorithms on the other hand, we apply the inhomogeneity and monotonicity conditions to the highway problem. We illustrate the setting by Example 3.3.2 below.

Example 3.3.2. *Figure 3.2 shows an example with three segments, $I = \{1, 2, 3\}$, and six customers $J = \{1, \dots, 6\}$. The left part of this figure shows the underlying highway with its alternative roads and costs for traveling, and the right part shows the corresponding instance of the highway problem.*

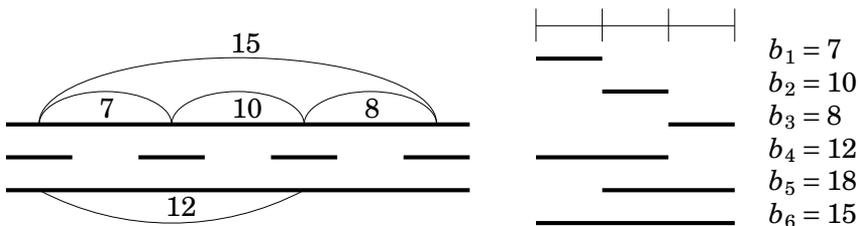


Figure 3.2: An instance of the highway problem.

There is no direct alternative road between the second and third segment, therefore, the valuation of customer 5 is equal to the sum of the costs for the

alternative roads of both segments separately. This instance has inhomogeneity $\alpha = 2$; comparing the average valuations for $\{1, 2, 3\}$ and $\{2\}$.

The approximation result derived in Section 3.2.2 holds for the single item pricing problem with arbitrary bundles. Obviously, it also holds for the highway problem.

Corollary 3.3.3. *Given $\varepsilon > 0$, the highway problem admits an approximation algorithm that yields a revenue at least $(1 + \ln \alpha + \varepsilon)^{-1}$ times the optimal revenue, with computation time $O(n(\log n + m))$.*

3.3.1 Complexity

Similarly as for the single item pricing problem with arbitrary bundles, the highway problem with homogeneous average valuations is trivially solvable. However, if we allow only arbitrarily small deviations of homogeneous valuations, the highway problem becomes intractable. In Theorem 3.3.6, we show that the highway problem with inhomogeneous average valuations remains NP-hard even if it is further restricted to Conditions 3.1, 3.2 and 3.3.

The reduction is from the PARTITION problem, and extends an idea by Briest and Krysta (2006). We use the PARTITION problem with an additional restriction, defined as follows.

Definition 3.3.4. *Given positive integers a_1, \dots, a_{2L} and A , does there exist a set $S \subseteq \{1, \dots, 2L\}$ such that $\sum_{\ell \in S} a_\ell = \sum_{\ell \notin S} a_\ell = A$ and $|S| = L$?*

This problem is known to be NP-hard; see Garey and Johnson (1979). We may assume that $L > 3/\varepsilon$, for otherwise the number of integers is bounded by a constant and therefore PARTITION is solvable in polynomial time. Without loss of generality, we also assume that $0 \leq a_1 \leq \dots \leq a_{2L}$ and $a_\ell \leq A$ for all $\ell = 1, \dots, 2L$. Let $a'_\ell = a_\ell + (4L + 2)A$ for all $\ell = 1, \dots, 2L$, and $A' = (4L^2 + 2L + 1)A$. Note that $\sum_{\ell=1}^{2L} a'_\ell = 2A'$.

We create an instance \mathcal{H} of the highway problem with $7L + 3$ segments. The segments are partitioned in $2L + 1$ gadgets. Gadget $\ell = 1, \dots, 2L$ consists of two segments, $i = 2\ell - 1$ and $i = 2\ell$. Each of these two segments are requested by $2L - 1$ customers with valuation a'_ℓ . The combination of two segments, $2\ell - 1$ and 2ℓ , is requested by one customer with valuation $(2 - \frac{1}{L})a'_\ell$. Gadget $2L + 1$ contains $3L + 3$ segments, where the first three segments, $4L + 1, 4L + 2, 4L + 3$, are requested by one customer with valuation $\frac{12}{4L+3}A'$ and the last $3L$ segments, $4L + 4, \dots, 7L + 3$, are requested by three

customers with valuation $\frac{12L}{4L+3}A'$. Three customers request all segments in gadget $2L + 1$ and each has valuation $\frac{12L}{4L+3}A'$. Finally, there is one *big* customer, who requests the first $4L + 3$ items with valuation $(4 - \frac{1}{L})A'$. Instance \mathcal{H} is displayed in Figure 3.3, where the number of customers interested in a subpath is presented by d .

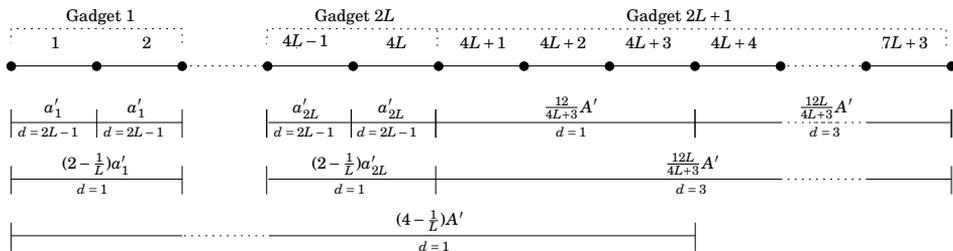


Figure 3.3: Instance \mathcal{H} .

In the following lemma, we show that this instance of the highway problem satisfies the three above mentioned conditions, that is, the inhomogeneity of average valuations is bounded from above by $1 + \varepsilon$ for some constant $\varepsilon > 0$, the valuations are monotone, and the average valuations are monotone decreasing. Then, using this lemma, we prove that the problem is NP-hard.

Lemma 3.3.5. *Instance \mathcal{H} of the highway problem satisfies Conditions 3.1, 3.2 and 3.3.*

Proof. The customers requesting item $4L - 1$ or $4L$ have the largest average valuation of a'_{2L} , and the customers requesting bundle $\{4L + 1, \dots, 7L + 3\}$ have the lowest average valuation of $\frac{12L}{12L^2+21L+9}A'$. Then for Condition 3.1 we have

$$\begin{aligned} \alpha &= \frac{a'_{2L}}{\frac{12L}{12L^2+21L+9}A'} = \frac{12L^2+21L+9}{12L} \cdot \frac{a'_{2L}}{A'} = \left(L + \frac{21}{12} + \frac{3}{4L}\right) \frac{a_{2L} + (4L+2)A}{(4L^2+2L+1)A} \\ &\leq \left(L + \frac{21}{12} + \frac{3}{4L}\right) \frac{4L+3}{4L^2+2L+1} = \frac{4L^2+10L+\frac{33}{4}+\frac{9}{4L}}{4L^2+2L+1} \\ &< 1 + \frac{2}{L} + \frac{13}{16L^2} < 1 + \frac{3}{L} < 1 + \varepsilon. \end{aligned}$$

Condition 3.2 follows straightforwardly by just verifying that the valuation on a bundle is at least as high as the valuation on any of its subsets. Now, it is easy to see that Condition 3.3 holds for the subpaths containing

one or two items. Also, for the customers requesting $\{4L+4, \dots, 7L+3\}$ and $\{4L+1, \dots, 7L+3\}$ it is obvious as their valuations are equal but the size of their bundles differ. We perform a case study to verify that Condition 3.3 also holds for all other customers $j, j' \in J$ for which $I_j \subseteq I_{j'}$. The average valuation of the bundles of size 1 is greater than the average valuation of the bundles of size 2, for all $\ell = 1, \dots, 2L$, thus for the single item bundles the condition holds trivially. If $I_j = \{2\ell - 1, 2\ell\}$ for $1 \leq \ell \leq 2L$ and $I_{j'} = \{1, \dots, 4L+3\}$ then we have

$$\begin{aligned} \left(1 - \frac{1}{2L}\right) a'_\ell &> \left(1 - \frac{1}{2L}\right) (4L+2)A = \frac{4L-1}{4L^2+3L} \left(4L^2+4L - \frac{3}{4L-1}\right)A \\ &> \frac{4L-1}{4L^2+3L} A'. \end{aligned}$$

For $I_j = \{4L+1, \dots, 4L+3\}$ and $I_{j'} = \{1, \dots, 4L+3\}$ we have

$$\frac{4}{4L+3} A' = \frac{4L}{4L^2+3L} A' > \frac{4L-1}{4L^2+3L} A'.$$

Finally, for $I_j = \{4L+1, \dots, 4L+3\}$ and $I_{j'} = \{4L+1, \dots, 7L+3\}$ we have

$$\frac{4}{4L+3} A' = \frac{12L}{12L^2+9L} A' > \frac{12L}{12L^2+21L+9} A'.$$

Consequently, instance \mathcal{H} of the highway problem satisfies all three conditions. \square

Theorem 3.3.6. *The highway problem is NP-hard even when restricted to Conditions 3.1, 3.2 and 3.3.*

Proof. We claim that there exists a feasible solution to PARTITION (see Definition 3.3.4) if and only if there is a feasible solution to instance \mathcal{H} of the highway problem with a total revenue of at least $(8L + \frac{72L}{4L+3} - \frac{1}{L})A'$.

Given a set $S \subseteq \{1, \dots, 2L\}$ such that $\sum_{\ell \in S} a_\ell = \sum_{\ell \notin S} a_\ell = A$ and $|S| = L$. For all $\ell \in \{1, \dots, 2L\}$, let $p_{2\ell-1} = p_{2\ell} = a'_\ell$ if $\ell \in S$ and $p_{2\ell-1} = p_{2\ell} = (1 - \frac{1}{2L})a'_\ell$ if $\ell \notin S$. Furthermore, we set $p_{4L+1} = \dots = p_{7L+2} = 0$ and $p_{7L+3} = \frac{12L}{4L+3}A'$. Applying this price vector, the revenue without contribution of the big customer is equal to $(4L-2)a'_\ell$ in each gadget $\ell = 1, \dots, 2L$. The big customer contributes her entire valuation $(4 - \frac{1}{L})A'$. In gadget $2L+1$, the customer requesting segments $4L+1, 4L+2$ and $4L+3$ gets this path for free.

The other customers in this gadget contribute their complete valuations. The total revenue generated with this pricing vector equals

$$(4L - 2) \sum_{\ell=1}^{2L} a'_\ell + \left(4 - \frac{1}{L}\right) A' + 6 \cdot \frac{12L}{4L+3} A' = \left(8L + \frac{72L}{4L+3} - \frac{1}{L}\right) A'.$$

For the converse, we are given an optimal solution to instance \mathcal{H} with a total revenue of at least $(8L + \frac{72L}{4L+3} - \frac{1}{L})A'$. First, we observe that in such an optimal solution, segments $4L+1$, $4L+2$ and $4L+3$ are necessarily priced to 0 and the total price of the remaining segments in gadget $2L+1$ is $\frac{12L}{4L+3}A'$, yielding a revenue of $\frac{72L}{4L+3}A'$. To see this, we notice that the total demand on the first three segments in this gadget is 5 and on the latter $3L$ segments the demand is 6. Therefore, if the total price on the first three segments of gadget $2L+1$ is $0 < x \leq \frac{12}{4L+3}A'$, the total revenue obtained in the gadget is at most $\frac{72L}{4L+3}A' - x$. That is, we receive x from the big customer and at most

$$x + 3 \left(\frac{12L}{4L+3} A' - x \right) + 3 \left(\frac{12L}{4L+3} A' \right)$$

from the customers in gadget $2L+1$. The above suggested pricing, where segments $4L+1$, $4L+2$ and $4L+3$ are priced to 0, does not decrease revenue generated in gadgets $1, \dots, 2L$, and the total revenue in gadget $2L+1$ is equal to $\frac{72L}{4L+3}A'$.

Second, in the optimal solution to the highway problem, there could be only two alternative pricing strategies in gadgets $\ell = 1, \dots, 2L$:

$$\text{either } p_{2\ell-1} = p_{2\ell} = a'_\ell \text{ or } p_{2\ell-1} + p_{2\ell} = \left(2 - \frac{1}{L}\right) a'_\ell,$$

where both prices do not exceed a'_ℓ . In both pricings, the contribution of the gadget (without big customer) to the total revenue is $(4L-2)a'_\ell$. Therefore, in the optimal solution to instance \mathcal{H} with revenue at least $(8L + \frac{72L}{4L+3} - \frac{1}{L})A'$, the big customer must contribute at least

$$\left(8L + \frac{72L}{4L+3} - \frac{1}{L}\right) A' - \frac{72L}{4L+3} A' - \sum_{\ell=1}^{2L} (4L-2) a'_\ell = \left(4 - \frac{1}{L}\right) A',$$

that is, at least her valuation. As she cannot spend more, she spends her total valuation. This amount is to be spent on the first $4L$ segments as the price of segments $4L+1$, $4L+2$ and $4L+3$ is set to 0.

Define set $S = \{\ell \in \{1, \dots, 2L\} : p_{2\ell-1} = p_{2\ell} = a'_\ell\}$. The payment of the big customer is

$$\sum_{\ell \in S} 2a'_\ell + \sum_{\ell \notin S} \left(2 - \frac{1}{L}\right) a'_\ell = \sum_{\ell=1}^{2L} \left(2 - \frac{1}{L}\right) a'_\ell + \sum_{\ell \in S} \frac{1}{L} a'_\ell = \left(4 - \frac{2}{L}\right) A' + \sum_{\ell \in S} \frac{1}{L} a'_\ell.$$

As this must be equal to the valuation of the big customer, we know that $\sum_{\ell \in S} a'_\ell = A'$. Also, as $\sum_{\ell=1}^{2L} a'_\ell = 2A'$, we have $\sum_{\ell \notin S} a'_\ell = A'$. Rewriting the first equality gives

$$\begin{aligned} \sum_{\ell \in S} a'_\ell = A' &\Leftrightarrow \sum_{\ell \in S} (a_\ell + (4L + 2)A) = (4L^2 + 2L + 1)A \\ &\Leftrightarrow \sum_{\ell \in S} a_\ell = (4L^2 + 2L + 1)A - |S|(4L + 2)A \\ &= A + (L - |S|)(4L + 2)A. \end{aligned}$$

We claim that $|S| = L$. To prove this, suppose it is not true. First, assume that $|S| < L$, that is, $|S| \leq L - 1$. Then we have

$$\sum_{\ell \in S} a_\ell = A + (L - |S|)(4L + 2)A \geq A + (4L + 2)A > 2A,$$

which is not possible as $\sum_{\ell=1}^{2L} a_\ell = 2A$, so $|S| \geq L$. Now, assume $|S| > L$, that is, $|S| \geq L + 1$. Then we have

$$\sum_{\ell \in S} a_\ell = A + (L - |S|)(4L + 2)A \leq A - (4L + 2)A < 0.$$

This is also not possible as all integers a_ℓ are nonnegative. Therefore, we can conclude that $|S| = L$ and consequently, $\sum_{\ell \in S} a_\ell = \sum_{\ell \notin S} a_\ell = A$. \square

3.4 Conclusion

As a main contribution of this chapter, we show that the highway problem as well as the single item pricing problem with arbitrary bundles is NP-hard (weakly and strongly, respectively) if the inhomogeneity is bounded from above by $1 + \varepsilon$, for any $\varepsilon > 0$. As it is easy to show that both problems are straightforwardly solvable if all average valuations are equal, we delineate the borderline between triviality and NP-hardness.

Unfortunately, the study of the natural version of the highway pricing problem in this chapter did not lead to closing the gap between logarithmic

approximation and weak NP-hardness. However, for instances where the average valuation between customers does not vary a lot, we present a good approximation algorithm. Clearly, the existence of a quasi-PTAS for the highway pricing problem by Elbassioni, Sitters, and Zhang (2007) suggests that a PTAS might be in reach. Yet, we leave it as an open problem to derive a PTAS, even for bounded inhomogeneity of valuations.

Chapter 4

Pricing in Capacitated Graphs

In this chapter we address the algorithmic complexity of a single item pricing problem in capacitated, undirected graphs. A company wants to determine the prices of capacitated edges to serve a set of potential customers. Each customer is interested in purchasing a simple path in the graph and has a maximum valuation that we assume to be known to the company. The goal is to decide which customers to serve, and to determine prices for all edges in order to maximize the company's total revenue. We address this single item pricing problem in different graph topologies. More specifically, we derive several results on the algorithmic complexity of this single item pricing problem, given that the graph is either a path, cycle, tree, or grid.¹

4.1 Introduction

We consider a single item pricing problem that is defined on a capacitated, undirected graph, referred to as the tollbooth problem and introduced by Guruswami et al. (2005). Given is a simple undirected graph $G = (V, E)$ with $|E| = m$ edges, and given are integral edge capacities c_e , $e \in E$. Each edge can be thought of as a network link, and the edge capacity determines the maximum number of customers the link can accommodate. We mainly, but not exclusively, discuss problems where edge capacities c_e are finite, in contrast

¹This chapter is based on Grigoriev, van Loon, Sitters, and Uetz (2006) and Grigoriev, van Loon, Sitters, and Uetz (2009).

to the original tollbooth problem. Given is a set of n potential customers $J = \{1, \dots, n\}$ each of which is interested in purchasing a network connection between two vertices of the graph. In contrast to many classical network problems, we assume that each customer wants one specific simple path, her bundle, denoted $E_j \subseteq E$, rather than *any* path that connects the two vertices. Each *single-minded* customer $j \in J$ has an integral *valuation* b_j , which is the largest amount that a customer is willing to pay for her path E_j . Let $B = \max_{j \in J} \{b_j\}$ be an upper bound on the valuations. The company determines prices for all edges, and the price of path E_j is equal to the sum of the prices of all edges in this path. A customer is a *winner* if the total price of her requested path does not exceed her valuation. The set of winners is denoted by $W \subseteq J$.

Definition 4.1.1. *A feasible solution consists of a set of winners $W \subseteq J$ and a vector of prices $p = (p_1, \dots, p_m) \in \mathbb{R}_+^m$, such that each edge $e \in E$ accommodates no more than c_e winners, and all winners can afford their path, that is, $\sum_{e \in E_j} p_e \leq b_j$ for all $j \in W$. The optimization problem consists of finding a feasible solution such that the total revenue $\Pi(W, p) = \sum_{j \in W} \sum_{e \in E_j} p_e$ is maximized.*

One usually distinguishes between solutions that are *envy-free* and those that are not. In the setting with single-minded customers considered here, envy-freeness requires that if a customer j is *not* a winner then the total price of path E_j must exceed her valuation. However, we mostly address problems without requiring envy-freeness in this chapter. In Example 4.1.2, we present a situation where the underlying graph is a path, customers request a subpath and an optimal solution is not required to be envy-free.

Example 4.1.2. *Consider the rental of a set of similar houses in holiday parks on the basis of weekends and midweeks, that is, non-overlapping discrete periods. Customers are interested in renting a house in consecutive periods. Since all houses are identical, in any time period they need to have the same price. But prices may vary from period to period. We can thus interpret a time period as an edge in a path, with edge capacity equal to the number of houses available in that period. Notice that envy-freeness is not necessarily an issue, since the owner of the houses can freely decide to whom to rent them.*

In Section 4.2, we address the problem where the underlying graph G is a path, known as the *highway problem*. This problem is introduced by Guruswami et al. (2005) and motivated by setting tolls on a single highway. If

the edge capacities are unlimited, we derive an exact dynamic programming algorithm for the case of uniform valuations, leading to a logarithmic approximation algorithm for the problem with non-uniform valuations. Then, we regard the problem on a path with an upper bound C on the capacity of any edge, that is, $c_e \leq C$ for all $e \in E$. We propose a dynamic programming algorithm that computes an optimal solution in time $O(n^{2C} B^{2C} m)$. Based on our dynamic programming algorithm, we moreover derive a fully polynomial time approximation scheme (FPTAS) for the highway problem, given that the maximum capacity of any edge, C , is a constant. In contrast to previous results in this direction (see e.g. Hartline and Koltun (2005), Balcan and Blum (2006) and Briest and Krysta (2006)), our FPTAS does neither require a constant number of edges in the subpaths, nor an upper bound on the valuations or that the bundles are a *laminar family*. The NP-hardness proof for the highway problem can easily be adapted to prove NP-hardness for the same problem on a cycle, as shown in Section 4.3. Moreover, in this section, we not only regard the problem on a cycle with limited edge capacities in which each customer $j \in J$ requests a path $E_j \subseteq E$, but also the problem where customers only specify two vertices to be connected. For both problems, we show how to adapt the dynamic programming approach of Section 4.2.3 in order to solve them to optimality in time $O(n^{3C} B^{3C} m)$ and $O(n^{3C} B^{3C} 4^C m)$, respectively.

In Section 4.4 we address the problem where the capacity of any edge is exactly *one*. For the case that graph G is a path (highway problem), the problem reduces to finding a maximum weight independent set in an interval graph, which is polynomially solvable (Möhring 1985). When we generalize from a path to a cycle, the problem reduces to finding a maximum weight independent set in a circular arc graph, which is also known to be solvable in polynomial time (Golombic and Hammer 1988). Furthermore, if the underlying graph G is a tree, we can show that the problem remains polynomially solvable. When the underlying graph G is a grid, however, we show that it is NP-hard to approximate the maximum revenue within a factor $n^{1-\varepsilon}$, for any $\varepsilon > 0$. Recall that n is the number of customers.

4.2 Highway problem

In this section, we restrict the underlying graph $G = (V, E)$ to be a path, thus we address the highway problem. We first discuss the complexity of this problem. Thereafter, we present a dynamic programming algorithm for

the case with unlimited edge capacities and uniform valuations, which leads to a logarithmic approximation algorithm for the case with non-uniform valuations. Finally, we present a dynamic programming algorithm and a fully polynomial time approximation scheme for the case where edge capacities are bounded by some constant C .

4.2.1 Complexity

It is not hard to see that the highway problem is polynomially solvable if either the set of winners $W \subseteq J$ is given, or the vector of prices p is given.

Lemma 4.2.1. *The highway problem is polynomially solvable if the vector of prices $p = (p_1, \dots, p_m)$ is given.*

Proof. Given the vector of prices $p = (p_1, \dots, p_m)$, we only need to find a feasible set of winners that maximizes the total revenue. Whenever the edge capacities are unlimited, this is trivial and the set of winners is just $W = \{j \in J : \sum_{e \in E_j} p_e \leq b_j\}$.

For the case of limited capacity, let W' be the set of customers for whom the requested path is affordable, given the price vector p . For any edge e , we can not accommodate more than c_e customers. Let a_{ej} be equal to 1 if edge $e \in E_j$ for customer j , and 0 otherwise. We find a revenue-maximizing feasible subset of winners by solving linear program (4.1), where $x_j = 1$ if and only if customer j is a winner. Note that $x_j = 0$ for all customers $j \in J \setminus W'$.

$$\begin{aligned}
 \max \quad & \sum_{j \in W'} \left(\sum_{e \in E_j} p_e \right) x_j \\
 \text{s.t.} \quad & \sum_{j \in W'} a_{ej} x_j \leq c_e \quad \forall e \in E \\
 & 0 \leq x_j \leq 1 \quad \forall j \in W'
 \end{aligned} \tag{4.1}$$

The constraint matrix $\{a_{ej}\}_{e \in E, j \in J}$ of this linear program has the consecutive ones property, which was introduced by Fulkerson and Gross (1965). A matrix has the consecutive ones property if there is a permutation of its rows that places the 1's consecutively in every column. The constraint matrix in linear program (4.1) has this property because the requested path E_j of any customer j consists only of consecutive edges. A consecutive ones matrix is totally unimodular (Nemhauser and Wolsey 1988), which means that every square non-singular matrix has determinant -1 or +1. Hence, the corresponding polyhedron only has integral vertices (Hoffman and Kruskal

1956; Papadimitriou and Steiglitz 1998). Therefore, linear program (4.1) yields an integral optimal solution, and $W = \{j \in J : x_j = 1\}$. \square

If a feasible set of winners $W \subseteq J$ is given, we find an optimal price vector $p = (p_1, \dots, p_m)$ by solving linear program (4.2). In the case with limited edge capacities, the maximum capacity of the edges is taken into account.

$$\begin{aligned}
 \max \quad & \sum_{j \in W} \sum_{e \in E_j} p_e \\
 \text{s.t.} \quad & \sum_{e \in E_j} p_e \leq b_j \quad \forall j \in W \\
 & p_e \geq 0 \quad \forall e \in E
 \end{aligned} \tag{4.2}$$

Since this constraint matrix has the consecutive ones property, too, we obtain the following.

Lemma 4.2.2 (Guruswami et al. (2005, Lemma 5.1)). *The highway problem is polynomially solvable if a feasible set of winners $W \subseteq J$ is given. Moreover, if the valuations b_j are integral, there exists an optimal, integral price vector.*

Briest and Krysta (2006) show that the highway problem is NP-hard. For their reduction from PARTITION it suffices, but it is also necessary, that the capacity of some edge is at least 3. If the capacity of any edge is at most 2, the complexity status of the problem remains open.

4.2.2 Approximation algorithm

Consider the problem with unlimited edge capacities, referred to as the highway problem by Guruswami et al. (2005). First let us assume that every customer has the same positive valuation b , that is, all customers have *uniform* valuations.

Theorem 4.2.3. *The highway problem can be solved in $O(m^3n)$ time if it has unlimited edge capacities and uniform positive valuations.*

Proof. Consider the problem with $b_j = 1$ for all $j \in J$. There exists an optimal solution in which all prices are 0 or 1, because given a set of winners W in an optimal solution, the optimal prices can be found by solving linear program (4.2) with an integral vector (b_1, \dots, b_n) , and the constraint matrix satisfies the consecutive ones property. We obtain this solution using a polynomial time dynamic program.

Without loss of generality, we assume that the edges on the path (highway) are indexed consecutively. In the dynamic programming algorithm for the problem with $b_j = 1$, for every customer $j \in J$, let s_j denote the edge in path E_j with the smallest index and t_j the edge in E_j with the largest index. For all edges $k, \ell \in E$ with $k < \ell$, let $R(k, \ell)$ be the total optimal revenue obtained by the customers for whom $t_j \leq \ell$, given that $p_k = p_\ell = 1$ and $p_{k+1} = \dots = p_{\ell-1} = 0$. Furthermore, let $R(k, m+1)$ be the total optimal revenue obtained by all customers given that $p_k = 1$ and, if $k < m$, $p_{k+1} = \dots = p_m = 0$. We define $R(0, \ell) = |\{j \in J : t_j = \ell\}|$ for all $\ell = 1, \dots, m$. Then, given $R(e, k)$ for all $e = 0, \dots, k-1$ and $k < \ell \leq m+1$, we compute the revenue $R(k, \ell)$ by finding the maximum sum of $R(e, k)$ and the contribution of all customers $j \in J$ whose path E_j contains either edge k or ℓ (not both), but not edge e . Formally,

$$R(k, \ell) = \max_{0 \leq e < k} \left\{ R(e, k) + \sum_{j \in J} r_j(e, k, \ell) \right\}, \text{ where}$$

$$r_j(e, k, \ell) = \begin{cases} 1 & \text{if } e < s_j \leq k \quad \text{and} \quad k < t_j < \ell \\ 1 & \text{if } k < s_j \quad \text{and} \quad t_j = \ell \\ 0 & \text{otherwise.} \end{cases}$$

The optimal revenue is equal to $\max_{k \in E} \{R(k, m+1)\}$. The computation time of this dynamic program is $O(m^3 n)$. Note that Guruswami et al. (2005) introduce yet another $O(B^{B+2} n^{B+3})$ time dynamic program to solve the same problem with a constant upper bound B on the valuations. This complexity is $O(n^4)$ in case $b_j = 1$ for all $j \in J$.

After we solve the problem with $b_j = 1$ for all $j \in J$, we multiply p_e by b for every edge $e \in E$, to obtain the optimal solution for the problem with original valuation b . \square

If we consider the problem in which every customer's valuation b_j is in $[b, \delta b]$ for some $\delta > 1$, then we can round down all valuations to b and lose at most a factor of δ in the optimal revenue. Therefore, the following corollary follows straightforwardly from Theorem 4.2.3.

Corollary 4.2.4. *Consider the highway problem with unlimited edge capacities. If for every customer $j \in J$, the valuation b_j is such that $b \leq b_j \leq \delta b$ for $b \geq 0$ and $\delta \geq 1$, then there exists a polynomial time δ -approximation.*

For the same problem with arbitrary valuations, we can derive a polynomial time $O(\ln B)$ -approximation algorithm. To achieve this, let us define

subproblems such that the ℓ^{th} subproblem contains customers $\{j \in J : \delta^{\ell-1} \leq b_j < \delta^\ell\}$. Using Corollary 4.2.4, we find a δ -approximate solution for each of these $\log_\delta B$ subproblems and we derive a $(\delta \log_\delta B)$ -approximation algorithm for the highway problem with unlimited edge capacities. The approximation ratio $\delta \log_\delta B$ is minimized by letting $\delta = e$, and we arrive at the following corollary.

Corollary 4.2.5. *There exists an $(e \ln B)$ -approximation algorithm for the highway problem with unlimited edge capacities.*

In the remainder of this section, we consider the highway problem with *limited* edge capacities. That is, each edge $e \in E$ can accommodate no more than c_e customers. Let $C \geq \max_{e \in E} c_e$ be an upper bound on the capacity of any edge. We show that we can solve this problem in time $O(n^{2C} B^{2C} m)$ by finding a longest path in an acyclic digraph. Afterwards, we derive a fully polynomial time approximation scheme for this problem.

4.2.3 Dynamic program

We create an m -layered digraph D with an additional source s and sink t , layers 0 and $m+1$, respectively. There are arcs only from layer e to $e+1$, for $e = 0, \dots, m$. Hence, in any $s-t$ path, there are exactly $m+2$ nodes. In every node in layer e , corresponding to edge e , we store all winners j that are accommodated by edge e . Moreover, we store the respective total amounts all these winners spend on all edges (network links) in their respective path E_j up to and including edge e . For any node x , the path $s-x$ in the digraph represents a feasible partial solution. Arcs from node x of layer e to node y of layer $e+1$ are only introduced if the path $s-y$ represents a feasible extension of the partial solution represented by the path $s-x$. The length of an arc that connects a node of layer e to a node of layer $e+1$ is equal to the revenue earned on edge $e+1$, that is, the total amount that the corresponding winners pay for edge $e+1$. Therefore, the length of the longest $s-t$ path in digraph D is equal to the maximum total revenue. Moreover, the set of winners can be reconstructed from the longest $s-t$ path. Algorithm 4.1 shows a more formal description.

Theorem 4.2.6. *The dynamic programming algorithm solves the highway problem in $O(n^{2C} B^{2C} m)$ time, where C is an upper bound on the edge capacities.*

Algorithm 4.1: Dynamic Programming Algorithm

Input: The highway problem with maximum capacity of any edge at most C

Output: Accommodation of customers to edges and edge prices p_e

begin (construction of digraph D)

nodes: For each edge $e \in E$, we introduce a layer of nodes:

Denote by J^e the set of customers with $e \in E_j$;

By $K^e = (j_1, j_2, \dots, j_k)$ we denote any (sorted) subset of J^e of cardinality k , where $k \leq \min\{c_e, |J^e|\} \leq C$;

Define $H_j := \{0, 1, \dots, b_j\}$ as the possible total amount customer $j \in K^e$ can spend on edges $\{1, \dots, e\} \cap E_j$;

Let $h^e \in H_{j_1} \times H_{j_2} \times \dots \times H_{j_k}$ be a vector denoting how much each customer j spends on edges $\{1, \dots, e\} \cap E_j$, for each $j \in K^e$. If $K^e = \emptyset$, we let $h^e = 0$;

Let all such pairs (K^e, h^e) be the nodes in layer e of D , for $e = 1, \dots, m$;

Let s and t denote source and sink. To unify notation, assume $s = (\emptyset, 0)$ and $t = (\emptyset, 0)$;

arcs: Insert an arc a^e from node (K^{e-1}, h^{e-1}) to node (K^e, h^e) , for $e = 2, \dots, m$, if:

(1) For all $j \in K^{e-1}$ with $e \in E_j$, $j \in K^e$, and for all $j \in J^{e-1} \setminus K^{e-1}$ with $e \in E_j$, $j \notin K^e$.

(2) There exists a unique integral value $d \geq 0$ such that $d = h_j^e - h_j^{e-1}$ for all $j \in K^{e-1} \cap K^e$, and $d = h_j^e$ for all $j \in K^e \setminus K^{e-1}$.

We furthermore connect source node s to all nodes (K^1, h^1) , and we connect all nodes (K^m, h^m) to sink node t .

arc lengths: For an arc a^e that connects (K^{e-1}, h^{e-1}) and (K^e, h^e) : If $K^e = \emptyset$, we let the length of arc a^e be $\ell(a^e) = 0$, and if $K^e \neq \emptyset$, we let the length of arc a^e be $\ell(a^e) = d|K^e|$, where d is the (unique) value from condition (2) above.

end

Solution: Compute the longest $s - t$ path \mathcal{P} in digraph D . Whenever for customer j we have that $j \in K^e$ with $(K^e, h^e) \in \mathcal{P}$, edge e accommodates customer j . The price p_e for edge e equals $\ell(a^e)/|K^e|$, where a^e is the arc from path \mathcal{P} that connects nodes (K^{e-1}, h^{e-1}) and (K^e, h^e) .

Proof. Consider an arbitrary $s - t$ path \mathcal{P} in digraph D . Abusing notation, let (W^e, h^e) be the nodes on \mathcal{P} . Set W^e is thus the set of customers that are accommodated by edge e , and h^e is the vector of the total amounts these customers pay up to and including edge e . By definition of the nodes and arcs of the digraph, no customer j will be accommodated by an edge outside her requested path E_j . For any customer j , consider an edge $e \in E_j$ such that $j \in W^e$. That is, edge e accommodates customer j . By condition (1) of the arc definition, all other edges of path E_j must accommodate customer j as well. Next, by definition we have for any node (W^e, h^e) that $|W^e| \leq c_e$. Hence, the edge capacities are satisfied. Finally, let us consider the valuation constraint of customer j . We know that $E_j = \{k, \dots, l\}$ for some edges $k \leq l$. We have that

$$\sum_{e \in E_j} p_e = \sum_{e=k}^l p_e = \sum_{e=k}^l \frac{\ell(a^e)}{|W^e|} = h_j^k + \sum_{e=k+1}^l (h_j^e - h_j^{e-1}) = h_j^l \leq b_j.$$

The third equality holds due to condition (2) of the arc definition, and the last inequality holds because $h_j^l \in H_j = \{0, 1, \dots, b_j\}$.

Now we know that any $s - t$ path \mathcal{P} in D defines a feasible solution to the highway problem with limited edge capacities, and $W = \bigcup_{e \in E} W^e$ denotes the set of winners in that solution. The length of path \mathcal{P} is

$$\sum_{a^e \in \mathcal{P}} \ell(a^e) = \sum_{e \in E} p_e |W^e| = \sum_{j \in W} \sum_{e \in E_j} p_e.$$

In other words, the path length defines the revenue of the corresponding solution, thus the longest path yields an optimal solution.

To arrive at the computation time of $O(n^{2C} B^{2C} m)$, we estimate the size of digraph D . For every edge $e \in E$, there are at most $O(n^C)$ different sets K^e and at most $O(B^C)$ different vectors h^e . Thus, per edge $e \in E$, we have at most $O(n^C B^C)$ nodes (K^e, h^e) . For any $e \in E$, every node (K^e, h^e) is connected to at most $O(n^C B^C)$ nodes (K^{e+1}, h^{e+1}) . So, per layer e , there are at most $O(n^{2C} B^{2C})$ arcs to layer $e + 1$, which means that there are at most $O(n^{2C} B^{2C} m)$ arcs in D . The computation time to find the longest path in D is linear in the number of arcs, since D is acyclic (Ahuja, Magnanti, and Orlin 1993). \square

Notice that the solution constructed by the dynamic programming algorithm is not necessarily envy-free.

4.2.4 FPTAS

We next show how to turn the dynamic programming algorithm into a fully polynomial time approximation scheme (FPTAS); that is, for any $\varepsilon > 0$, we have an algorithm that computes a solution with revenue at least $(1 - \varepsilon)$ times the optimum revenue, in time polynomial in the input and $1/\varepsilon$. To that end, we just apply the dynamic programming algorithm to a rounded instance in which the customers' valuations are $b'_j = \lfloor b_j/\beta \rfloor$ where $\beta = \varepsilon B/(2n^2)$ for $\varepsilon > 0$. Note that without loss of generality, we may assume that β is integer.

Lemma 4.2.7. *For every solution (W, p) to the original instance, there exists a solution (W, p'') to the rounded instance with revenue $\Pi(W, p'') > \frac{1}{\beta}\Pi(W, p) - mn$.*

Proof. Let (W, p) be a feasible solution to the original instance with revenue $\Pi(W, p)$. Let $p''_e = \lfloor p_e/\beta \rfloor$ for all edges $e \in E$. Note that $(p_e/\beta) - 1 < p''_e \leq (p_e/\beta)$. For the original instance we have for every winner $j \in W$, $\sum_{e \in E_j} p_e \leq b_j$, and it follows that

$$\sum_{e \in E_j} p''_e = \sum_{e \in E_j} \left\lfloor \frac{p_e}{\beta} \right\rfloor \leq \left\lfloor \frac{\sum_{e \in E_j} p_e}{\beta} \right\rfloor \leq \left\lfloor \frac{b_j}{\beta} \right\rfloor = b'_j.$$

Hence, the solution (W, p'') is feasible to the rounded instance. Finally, we have

$$\Pi(W, p'') = \sum_{j \in W} \sum_{e \in E_j} p''_e > \sum_{j \in W} \sum_{e \in E_j} \left(\frac{p_e}{\beta} - 1 \right) \geq \frac{1}{\beta} \Pi(W, p) - mn.$$

□

Lemma 4.2.8. *For every solution (W', p') to the rounded instance, there exists a solution (W', \tilde{p}) to the original instance with $\Pi(W', \tilde{p}) = \beta \Pi(W', p')$.*

Proof. Let (W', p') be a solution to the rounded instance yielding revenue $\Pi(W', p')$. Let $\tilde{p}_e = p'_e \beta$ be prices in the original instance for all edges $e \in E$. This is integer because p'_e and β are integer. Then the valuation constraint for every customer $j \in W'$ is satisfied, because

$$\sum_{e \in E_j} \tilde{p}_e = \beta \sum_{e \in E_j} p'_e \leq \beta b'_j = \beta \left\lfloor \frac{b_j}{\beta} \right\rfloor \leq b_j.$$

Hence, the same set of customers W can get their requested path, and solution (W', \tilde{p}) is feasible to the original instance. The revenue can be written as

$$\Pi(W', \tilde{p}) = \sum_{j \in W'} \sum_{e \in E_j} \tilde{p}_e = \sum_{j \in W'} \sum_{e \in E_j} (p'_e \beta) = \beta \sum_{j \in W'} \sum_{e \in E_j} p'_e = \beta \Pi(W', p').$$

□

We now combine Lemma 4.2.7 and Lemma 4.2.8 to obtain an FPTAS.

Theorem 4.2.9. *There exists an FPTAS for the highway problem in which edge capacities are bounded by a constant.*

Proof. Let (W, p) and (W', p') be the optimal solutions to the original and rounded instances, respectively. Consider solution (W', \tilde{p}) to the original instance, where $\tilde{p}_e = \beta p'_e$ for all edges $e \in E$, and solution (W, p'') to the rounded instance, where $p''_e = \lfloor p_e / \beta \rfloor$ for all edges $e \in E$. An application of the previous two lemmas now yields

$$\begin{aligned} \Pi(W', \tilde{p}) = \beta \Pi(W', p') &\geq \beta \Pi(W, p'') \\ &> \beta \left(\frac{1}{\beta} \Pi(W, p) - mn \right) = \Pi(W, p) - \varepsilon B \frac{mn}{2n^2}, \end{aligned}$$

where the first inequality holds due to optimality of solution (W', p') to the rounded instance. Note that for n customers there are at most $2n$ distinct endpoints of their requested paths. Without loss of generality, we identify a set of edges between two consecutive endpoints with a single edge. Therefore, the number of distinct edges m is at most $2n - 1$. Since the maximum revenue $\Pi(W, p)$ is at least the maximum valuation B , we have that $\Pi(W', \tilde{p}) > (1 - \varepsilon) \Pi(W, p)$.

Concerning the time to compute the optimal solution (W', p') , observe that the upper bound on the valuation is now

$$\max_{j \in J} \{b'_j\} = \max_{j \in J} \left\{ \left\lfloor \frac{b_j(2n^2)}{\varepsilon B} \right\rfloor \right\} \leq \frac{B(2n^2)}{\varepsilon B} = \frac{2n^2}{\varepsilon}.$$

Thus, the number of nodes per layer $e \in E$ is at most $O(n^{3C}/\varepsilon^C)$ and consequently there are $O(n^{6C}/\varepsilon^{2C})$ arcs between any two consecutive layers. Now, as $m \leq 2n - 1$, the total number of arcs in the digraph is $O(n^{6C+1}/\varepsilon^{2C})$. As the computation time to find the longest path in an acyclic digraph D is linear in the number of arcs (Ahuja, Magnanti, and Orlin 1993), the computation time is polynomial in terms of n and $1/\varepsilon$. □

4.3 Pricing in a capacitated cycle

The complexity result and the dynamic program we obtained for the highway problem with capacitated edges can be extended to the pricing problem on a capacitated cycle.

Corollary 4.3.1. *The single item pricing problem on a cycle is NP-hard.*

In fact, the NP-hardness proof for the highway problem by Briest and Krysta (2006) immediately results in NP-hardness for the problem on a cycle, as the two ends of the path can be connected. Customers still request the same subpaths after this transformation.

Theorem 4.3.2. *The single item pricing problem on a capacitated cycle can be solved in time $O(n^{3C} B^{3C} m)$.*

Proof. We adapt the dynamic program of Algorithm 4.1. Choose any edge $e \in E$ to start the procedure. Other than in the previous algorithm we now start in layer 1, and include all possible nodes (K^e, h^e) in this layer; these nodes are created similarly as in Algorithm 4.1, where K^e is a subset of the customers for which $e \in E_j$, and h^e represents the total amount spent on edges up to and including edge e . For example, for a customer j with $E_j = \{e-2, e-1, e, e+1\}$, h^e represents the amount this customer spends on edges $e-2$, $e-1$ and e , regardless of the fact that we do not yet know how much exactly she spends on edges $e-2$ and $e-1$.

Arcs between layers e and $e+1$ exist if the conditions (1) and (2) of Algorithm 4.1 are satisfied. In the final layer, $m+1$, there are exactly the same nodes as in layer 1 to complete the cycle. For every node (K^e, h^e) in layer 1, we find the longest path to the *same* node (K^e, h^e) in layer $m+1$. Among all longest paths, we select the one with the highest total value. This value is equal to the maximum total revenue, and the set of winners can be reconstructed from this longest path as well.

The size of the digraph is similar as before, that is, there are $O(n^C B^C)$ nodes per layer, $O(n^{2C} B^{2C})$ arcs between any two consecutive layers, and therefore a total of $O(n^{2C} B^{2C} m)$ arcs. Finding a longest path in the digraph is linear in the number of arcs since the digraph is acyclic (Ahuja, Magnanti, and Orlin 1993). As we have to find the longest path for each of the $O(n^C B^C)$ nodes in the first layer, all longest paths can be found in $O(n^{3C} B^{3C} m)$, which is the total time needed to find the optimal solution to the problem on a cycle with limited edge capacities. \square

An interesting extension is the problem in which each customer requests a *connection* between two vertices instead of a specific path. In this case, the company has the freedom to decide in which direction of the cycle the connection is established in order to maximize the revenue.

Theorem 4.3.3. *The single item pricing problem on a capacitated cycle is NP-hard, even if customers are indifferent to the direction used for the connection.*

Proof. We reduce from the highway problem. First, we include an additional edge that connects the two end vertices of the original path to create a cycle. Then, we introduce one extra customer that requests a connection between the two original end vertices and has a valuation that is larger than the sum of all other customers' valuations. This forces the price on the new edge to be expensive enough in the optimal solution such that no other customer can afford using this edge. Therefore, all customers will still request the same paths as in the instance on a path. \square

To solve this problem, we further adapt the previous dynamic programming approach to include the direction of the connection between the requested vertices for every customer. Therefore, given set $K^e = (j_1, \dots, j_k)$ of customers with $e \in E_j$, we introduce vector $dir^e \in \{0, 1\}^k$ to denote for each customer $j \in K^e$ the direction of the connection between the requested vertices; 0 for clockwise, 1 for counter-clockwise. Thus, in layer 1, corresponding to chosen edge $e \in E$, we have all possible nodes (K^e, h^e, dir^e) .

Arcs are again created if conditions (1) and (2) of Algorithm 4.1 are met. However, yet another condition should be satisfied, which assures that the connection for each customer has only one direction. Thus, for all $j \in K^e \cap K^{e+1}$, there is an arc if also $dir_j^e = dir_j^{e+1}$. In the final layer, $m + 1$, there are again the same nodes as in layer 1. Finally, after the complete digraph is created, we find for every node (K^e, h^e, dir^e) in layer 1 the longest path to the corresponding node (K^e, h^e, dir^e) in layer $m + 1$. Among all longest paths, we select the one with the highest total value. This value is equal to the maximum total revenue, and the set of winners can be reconstructed from this longest path as well.

Theorem 4.3.4. *The single item pricing problem on a capacitated cycle in which customers are indifferent to the direction used for the connection can be solved in time $O(n^{3C} B^{3C} 4^C m)$.*

Proof. By a similar reasoning as in the proof of Theorem 4.3.2, and the extra condition that each customer that gets a connection between the requested vertices has a connection oriented in only one direction, the longest of all longest paths between similar nodes defines the revenue of the optimal solution.

For every edge $e \in E$, there are at most $O(n^C)$ different sets K^e , at most $O(B^C)$ different vectors h^e , and at most $O(2^C)$ different vectors dir^e . Thus, per edge $e \in E$, we have at most $O(n^C B^C 2^C)$ nodes (K^e, h^e, dir^e) . For any $e \in E$, every node (K^e, h^e, dir^e) is connected to at most $O(n^C B^C)$ nodes of the form $(K^{e+1}, h^{e+1}, dir^{e+1})$, as there only exists an arc when the values dir_j^e and dir_j^{e+1} are the same for all customers $j \in K^e \cap K^{e+1}$. So, there are at most $O(n^{2C} B^{2C} 2^C)$ arcs between layers e and $e + 1$, which means that there are at most $O(n^{2C} B^{2C} 2^C m)$ arcs in the digraph. The computation time to find the longest path in the digraph is linear in the number of arcs since the digraph is acyclic (Ahuja, Magnanti, and Orlin 1993). We have to find the longest path for each node in layer 1, which takes $O(n^{3C} B^{3C} 4^C m)$ time. \square

4.4 Unit supply

In this section we assume that the capacity of any edge is one, that is, $c_e = 1$ for all $e \in E$. We consider four types of graphs, namely paths, cycles, trees and grids.

4.4.1 Highway problem

The instance on a path can be represented as an interval graph. An *interval graph* is the intersection graph of a set of intervals on a path. It has one vertex for each interval, or subpath, in the instance, that is, one vertex per customer. There is an edge between every pair of vertices corresponding to intervals that intersect, that is, customers for which the intersection of their requested subpaths is not empty. The valuations of the customers are represented as weights on the corresponding vertices. As the edge capacities are one in this case, we are searching for a maximum weight independent set, that is, a subset of the vertices such that no two vertices in this subset are adjacent in the interval graph. The result of Theorem 4.4.1 is not surprising, since the problem of finding a maximum weight independent set in an interval graph is known to be solvable in quadratic time according to a result by Möhring (1985).

Theorem 4.4.1 (Rothkopf, Pekec, and Harstad (1998, Theorem 10)). *The highway problem with unit edge capacities can be solved in $O(n^2)$ time.*

4.4.2 Cycle

The instance on a cycle can be represented as a *circular arc graph*, which is the intersection graph of a set of intervals on a cycle. It has one vertex for each interval, or subpath, in the instance, and an edge between every pair of vertices corresponding to intervals that intersect. Again, the weight of a vertex is equal to the corresponding customer's valuation. Once more, we search for a maximum weight independent set in this circular arc graph, which is known to be solvable in cubic time according to a result by Golumbic and Hammer (1988).

Theorem 4.4.2 (Rothkopf, Pekec, and Harstad (1998, Corollary 11)). *The single item pricing problem on a cycle with unit edge capacities can be solved in $O(n^3)$ time.*

The problem in which the customers are indifferent to the direction of their requested connection can be solved by straightforward Algorithm 4.2, also in $O(n^3)$ time.

Algorithm 4.2: Single item pricing problem on a cycle with unit capacity and customers requesting a connection

Let \vec{E}_j be the edges contained in the clockwise connection for customer j , and \overleftarrow{E}_j the edges in the counterclockwise connection.

foreach $j = 1, \dots, n$ **do**

Assign \vec{E}_j to j , and solve the problem on path $E \setminus \vec{E}_j$ with customers $J \setminus j$. Let $\vec{\Pi}_j$ be the revenue;

Assign \overleftarrow{E}_j to j , and solve the problem on path $E \setminus \overleftarrow{E}_j$ with customers $J \setminus j$. Let $\overleftarrow{\Pi}_j$ be the revenue;

Let $\Pi_j = \max\{\vec{\Pi}_j, \overleftarrow{\Pi}_j\}$;

end

$\Pi = \max_{j \in J} \{\Pi_j\}$.

4.4.3 Tree

Guruswami et al. (2005) show that the problem with unlimited edge capacities is APX-hard even on star graphs. Contrasting this complexity result, we prove that if the capacity of each edge is exactly one, the problem on a tree can be solved in polynomial time. Again, recall that we do not require the solution to be envy-free.

Theorem 4.4.3. *The single item pricing problem on a tree can be solved in $O(n^4)$ time.*

Proof. Consider the graph $H = (J, E')$ where $(j, k) \in E'$ if and only if $E_j \cap E_k \neq \emptyset$, for two customers $j, k \in J$. Since $\{E_j : j \in J\}$ is a collection of simple paths in a tree, graph H is an edge intersection graph of paths in a tree, or *EPT graph*, as defined by Golubic and Jamison (1985); see also Brandstädt, Le, and Spinrad (1999). The underlying graph G is a tree and the capacity of each edge is exactly one, thus the maximum weight independent set in H with vertex weights $b_j, j \in J$, corresponds to the optimal set of winners W , and the weight of this independent set is equal to the maximum revenue. The vector of optimal prices can be obtained straightforwardly by setting the price of one arbitrary edge of E_j to b_j , and setting the prices of all other edges in E_j to 0, $j \in W$. The remaining edges in the tree can be priced arbitrarily.

A polynomial time algorithm to compute a maximum weight independent set in an EPT graph was described by Tarjan (1985). The algorithm is a recursive procedure that decomposes the problem on the basis of clique separators. The polynomial computation time is a consequence of the fact that the atoms, that is, the non-decomposable subgraphs of EPT graphs, are line graphs. A *line graph* of an undirected graph G is a graph that represents the adjacencies between edges of G . The line graph has a vertex for every edge in G . Two vertices of the line graph are adjacent if and only if their corresponding edges are adjacent in G . For line graphs, the maximum weight independent set problem is just the maximum weight matching problem, which can be solved in $O(n^3)$ time by Edmonds' algorithm (Edmonds 1965). As the total number of atoms is $O(n)$, the total time complexity is bounded by $O(n^4)$. \square

4.4.4 Grid

Demaine et al. (2006) show that the single item pricing problem with unlimited capacities, where the customers request arbitrary subsets of the network links, is hard to approximate within a semi-logarithmic factor. If we restrict the requested subsets of links to be a path of edges in a general graph, then the problem is APX-hard even under strong restrictions as shown by Guruswami et al. (2005) and Briest and Krysta (2006)

Here we show that if the capacities of the edges are bounded by one, then we can derive an even stronger inapproximability result for a very restricted class of graphs and customers' requests.

Theorem 4.4.4. *For all $\varepsilon > 0$, approximating the single item pricing problem on a grid with unit supply within a factor $n^{1-\varepsilon}$ is NP-hard, even with unit valuations, and when each edge is an element of at most two requested paths. The same result holds if the solution is required to be envy-free.*

Proof. We construct an approximation preserving reduction from INDEPENDENTSET. In the latter problem, given a graph $H = (V', E')$, the problem is to find a maximum cardinality subset $S \subseteq V'$ such that no two vertices from S are adjacent. It is NP-hard to approximate INDEPENDENTSET within a factor $|V'|^{1-\varepsilon}$ as shown by Zuckerman (2006).

Let $V' = \{v_1, \dots, v_n\}$ and $E' = \{a_1, \dots, a_m\}$. We construct the instance of the single item pricing problem as follows. We create an $(n+1) \times (2m+2)$ grid, that is, a simple graph with $2nm + 2n + 2m + 2$ vertices $\{(j, e) : 1 \leq j \leq n+1, 1 \leq e \leq 2m+2\}$, where (j, e) and (j', e') are adjacent if $|j' - j| + |e' - e| = 1$. Let horizontal layer $j \in \{1, \dots, n\}$ correspond to vertex $v_j \in V'$, and let the edge $((n+1, 2e), (n+1, 2e+1))$ in the grid correspond to edge $a_e \in E'$. Next, for each vertex $v_j \in V'$, we introduce a customer in the single item pricing problem with a requested path defined by the following simple path in the grid graph. The path starts at point $(j, 1)$ and ends at point $(j, 2m+2)$ following the layer j everywhere except for the edges $((j, 2e), (j, 2e+1))$ such that $v_j \in a_e$. These edges are substituted by vertical detours, passing through edges $((n+1, 2e), (n+1, 2e+1))$. See Figure 4.1 for an example. We complete the construction by setting the valuation of each customer to 1.

We claim that there exists an independent set of cardinality K in H if and only if there exists a solution to the single item pricing problem on the grid with unit edge capacities with total revenue K . By construction, two paths in the grid corresponding to adjacent vertices in H must share some edge $a \in E'$ in layer $(n+1)$. Since the capacity of edge a is 1, only one of these paths can

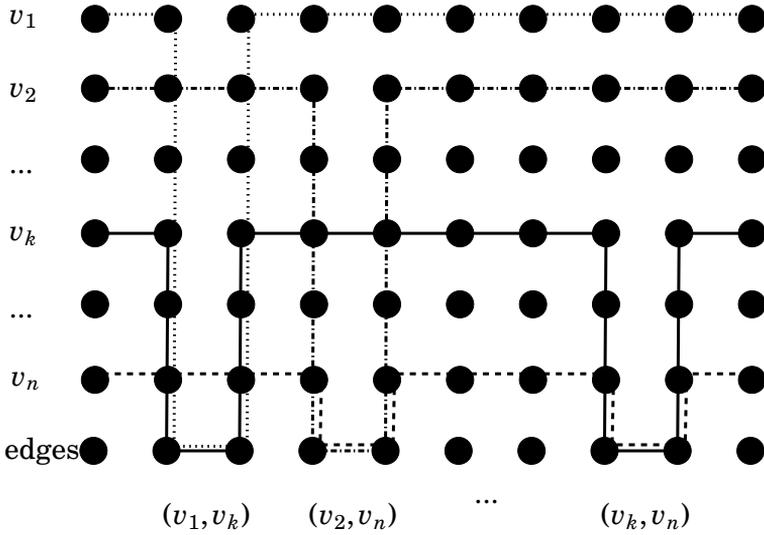


Figure 4.1: Grid graph from the reduction of INDEPENDENTSET.

be present in a feasible solution. Hence, the total revenue in the single item pricing problem is at most the maximum cardinality independent set in H .

For the converse, consider an independent set S in H and any two vertices in this independent set. By construction, the two corresponding paths in the single item pricing problem are edge disjoint. Therefore, there is a solution to the single item pricing problem where S defines the set of winners and each edge $e \in E$ accommodates at most one customer. For each $v_j \in S$ we set the price of the grid edge $((j, 1), (j, 2))$ to 1, for each $v_j \notin S$ we set the price of $((j, 1), (j, 2))$ to 2, and for all other edges of the grid we set the prices to 0. In the constructed solution to the single item pricing problem the total revenue equals $|S|$. Thus, the reduction preserves the objective value.

Since the number of customers n in the single item pricing problem exactly equals $|V'|$, we derive that the single item pricing problem is hard to approximate within a factor $n^{1-\epsilon}$. It remains to notice that the constructed solution is envy-free since the price of the bundle of each non-winning customer equals 2, which is greater than her valuation. Therefore, the theorem holds also if we require the solution to be envy-free. \square

4.5 Conclusion

For the highway problem with capacitated edges, we derive an FPTAS based on a pseudo-polynomial time dynamic program. These results are extended to the problem where the underlying graph is a cycle. For the problem where the capacity is equal to one, we show that the problem is polynomially solvable if the underlying graph is a path, cycle or tree. However, on a grid, the problem becomes difficult to approximate within $n^{1-\varepsilon}$ for any $\varepsilon > 0$.

Part II

Affine Pricing

Chapter 5

Algorithms for Affine Pricing

We consider a revenue maximization problem where a company sells a set of items to a set of customers. Every customer has one request for a contract based on the different types of items available. We assume that the requests and the valuations of the customers are known to the company. The company determines a price for every type of item. Given these prices, the price of a contract is calculated on the basis of an arbitrary affine function on the request of the customer. We show that this model generalizes the single item pricing problem, and we derive a fixed-parameter polynomial time algorithm for the affine pricing problem.¹

5.1 Introduction

A telephone operator offers various types of phone usage to its customers, for example, calling one minute domestically or abroad, sending a text message or multimedia objects, but also establishing a connection. The operator has to determine a price for each type of phone usage. The question is how to set the prices so as to maximize the total revenue. Through market research, an operator can acquire the exact details of phone usage for each of its customers, that is, data is available on the date and time of the usage, the duration, and the used network. Every customer requests a *contract* based on her specific phone usage. Given the prices for the different types of phone usage set by the operator, the price of the contract of a particular customer is an affine function on her request.

¹This chapter is based on Grigoriev, van Loon, Sviridenko, Uetz, and Vredeveld (2007).

Affine pricing is a pricing strategy where there is a fixed, customer dependent cost for accepting the contract, and a constant per-item price for every item type. Affine pricing is different from the single item pricing problem described in Part I, as there customers request subsets of the available set of items. Single item pricing is semi-logarithmic inapproximable, that is, hard to approximate within a factor of $O(\log^\epsilon n)$ for some $\epsilon > 0$, assuming that $\text{NP} \not\subseteq \text{BPTIME}(2^{O(n^\delta)})$ for all $\delta > 0$, as was generally shown by Demaine et al. (2006). In this chapter, we show that the affine pricing model generalizes the single item pricing problem and give complexity results in case the *dimension* of the affine pricing problem, that is, the number of distinct item types, is not constant. If the dimension of the problem is constant, we present a polynomial time algorithm to solve the problem. In Section 5.4, we apply the affine pricing algorithm in a practical study using data from a telephone company that resemble reality. Moreover, we present a local search algorithm that yields solutions very close to optimal for the practical data at hand.

The affine pricing problem is defined as follows. Let $J = \{1, \dots, n\}$ denote the set of customers requesting a contract from a company. Every customer $j \in J$ has a *valuation* b_j , which is the maximum amount j is willing to pay for the contract. The customers are *single-minded*, that means that a customer requests one contract which she accepts only if its price does not exceed her valuation. Let set $K = \{1, \dots, m\}$ be the set of distinct *item types*, each influencing the price of the contract, and we call m the dimension of the affine pricing problem. Note that an item is not necessarily a physical or digital product; it can for example also be a service or shipping cost. The price of item type $k \in K$ is denoted by $p_k \geq 0$. The price $p(j)$ of the contract requested by customer j is defined by affine price function

$$p(j) = d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m, \quad j \in J. \quad (5.1)$$

Positive coefficient d_{jk} represents the demand of customer $j \in J$ for item type $k \in K$. Coefficient d_{j0} is the cost charged to customer $j \in J$ for accepting the contract. Without loss of generality, we assume that $d_{j0} \leq b_j$. The coefficients may in general depend on the item types requested in the contract, but also on customer j or the requested contract itself. For example, if k represents one minute of calling then the number of minutes customer j wants to call is equal to d_{jk} . Or, if k represents shipping then d_{jk} is equal to 1 if customer j requests shipping and 0 otherwise. Thus every customer gets a *personal* offer for the contract she requests.

The problem consists of determining nonnegative prices p_k for the item types $k \in K$. A solution to the problem is a price $p(j)$ for the contract requested by customer $j \in J$. A customer is called a *winner* if she accepts and receives the contract, and a *loser* otherwise. The set of winners is denoted by $W = \{j \in J : p(j) \leq b_j\}$.

Definition 5.1.1. *A solution is feasible if the contracts of all winners can be afforded, that is, the price of the contract is at most the customer's valuation, and if no item type is oversold in case of limited supply. The objective of the company is to maximize the total revenue, $\Pi = \sum_{j \in W} p(j)$.*

Definition 5.1.2 (Envy-free). *A solution is envy-free if it is feasible and in addition the losers' contracts are each priced higher than their valuations.*

In this chapter, we only consider envy-free solutions. This is a property of an optimal solution to the pricing problem with unlimited supply of item types, but has to be assured if the supply is limited.

5.2 Affine pricing algorithm

We present an algorithm to solve the affine pricing problem. The running time of this algorithm is polynomial as long as the dimension of the affine pricing problem m is constant. We describe the algorithm with unlimited supply of the item types, and extend it to obtain a near-optimal solution for the case with limited supply in Section 5.2.1. In the final part of this section, we describe the unlimited supply problem with unit-demand customers instead of single-minded customers.

Definition 5.2.1 (Arrangement of linear inequalities). *An arrangement of linear inequalities is a set of linear inequalities in m dimensional space.*

Definition 5.2.2 (Vertex). *A vector $p = (p_1, \dots, p_m) \in \mathbb{R}_+^m$ is a vertex of the arrangement of valuation and nonnegativity inequalities if it is characterized by m out of the following $n + m$ equalities*

$$\begin{aligned} d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m &= b_j, \quad j \in J \\ p_k &= 0, \quad k \in K, \end{aligned}$$

such that the coefficient matrix of these m equalities has rank m .

The affine pricing algorithm that we propose in this chapter, enumerates all vertices of the arrangement of *valuation inequalities* $p(j) \leq b_j$, $j \in J$, and *nonnegativity inequalities* $p_k \geq 0$, $k \in K$. We select the vertex for which the price vector yields the highest revenue. We show in Lemma 5.2.3 that there exists a vertex that is an optimal solution to the affine pricing problem, which proves that enumerating all vertices suffices to solve the problem.

Lemma 5.2.3. *There exists a vertex that is an optimal pricing strategy to the affine pricing problem.*

Proof. Given an optimal solution $p^* = (p_1^*, \dots, p_m^*) \in \mathbb{R}_+^m$. Let $W^* = \{j \in J : p^*(j) \leq b_j\}$ be the set of winners. Consider the following linear program.

$$\begin{array}{ll} \max & (\sum_{j \in W^*} d_{j0}) + (\sum_{j \in W^*} d_{j1})p_1 + \dots + (\sum_{j \in W^*} d_{jm})p_m \\ \text{s.t.} & d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m \leq b_j \quad \forall j \in W^* \\ & p_k \geq 0 \quad \forall k \in K. \end{array}$$

Let $\bar{p} = (\bar{p}_1, \dots, \bar{p}_m) \in \mathbb{R}_+^m$ be an optimal solution to this linear program. Now, the generated revenue with vector \bar{p} is at least as high as the generated revenue with vector p^* , as \bar{p} is an optimal solution to the linear program and they both have the same set of winners. By optimality of p^* , the revenue should be equal for both price vectors. By definition, vector \bar{p} is a vertex. \square

Details of the affine pricing algorithm are presented in Algorithm 5.1. The algorithm is applied to a simple instance in Example 5.2.4 to further clarify the procedure.

Algorithm 5.1: Affine pricing algorithm.

```

Let  $\Pi^* = 0$ ,  $p^* = (0, \dots, 0)$  and  $W^* = J$ ;
foreach vertex  $p \in \mathbb{R}_+^m$  do
    Let  $p(j) = d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m$  be the price of the contract
    requested by customer  $j \in J$ ;
    Let  $W = \{j \in J : p(j) \leq b_j\}$  be the set of winners;
    Let  $\Pi = \sum_{j \in W} p(j)$  be the total revenue;
    if  $\Pi > \Pi^*$  then
        | Let  $\Pi^* = \Pi$ ,  $p^* = p$  and  $W^* = W$ ;
    end
end

```

Example 5.2.4. Assume the company has two different item types for sale, and three customers request a contract. The fixed cost $d_{j0} = 0$ for all customers j . One customer wants 4 copies of the first item, and 1 of the second. One customer is interested in 6 copies of the first and 1 of the second item. And one customer requests 1 copy of both items. Their valuations are 4, 3 and 2, respectively. The valuation inequalities are $4p_1 + p_2 \leq 4$, $6p_1 + p_2 \leq 3$ and $p_1 + p_2 \leq 2$. Figure 5.1 shows a graphical representation of this situation. The bullets represent the price vectors that are vertices. For each such price vector we determine the revenue Π . The optimal revenue of $7\frac{3}{5}$ is attained at $p = (\frac{1}{5}, 1\frac{4}{5})$. Graphically, we can see that $W = J$ because this price vector lies in the feasible region defined by the valuation inequalities $p(j) \leq b_j$ of all customers $j \in J$.

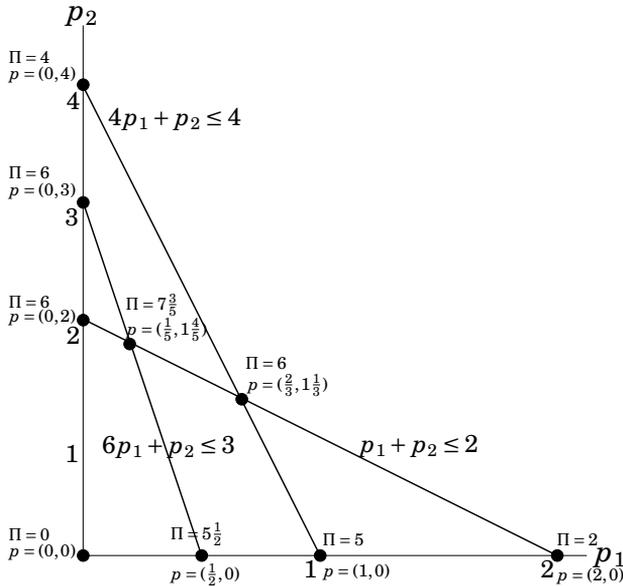


Figure 5.1: Example of an instance of affine pricing.

Theorem 5.2.5. The affine pricing problem can be solved in polynomial time if the number of item types m is constant. The computation time is $O((n + m)^m(m^3 + nm)) \in O(n^{m+1})$.

Proof. Correctness of Algorithm 5.1 immediately follows from Lemma 5.2.3. We consider $\binom{n+m}{m} \in O((n + m)^m)$ systems of m equalities each. In each of these iterations, we solve a linear system in m variables and m constraints

to determine the price vector, which takes $O(m^3)$ time. Checking whether all prices are positive takes $O(m)$ time. Computation of the contract prices, winners, and the revenue takes $O(nm)$ time. The claimed time complexity follows. \square

In contrast, if the dimension m of the affine pricing problem is not constant, the problem is much harder. In fact, if m is not constant, the single item pricing problem is just a special case of the affine pricing problem: For all $k \in K$ and $j \in J$, let $d_{jk} = 1$ whenever item type k is contained in j 's bundle and $d_{jk} = 0$ otherwise, and let $d_{j0} = 0$. Then, each item price corresponds to one variable p_k , thus the affine pricing problem generalizes the single item pricing problem. Combining this with a result by Demaine et al. (2006), we obtain the following.

Corollary 5.2.6. *The affine pricing problem with an unbounded number of item types is inapproximable within a semi-logarithmic factor.*

5.2.1 Limited supply

In this section, we consider the envy-free affine pricing problem with limited availability of item types. Let $c_k \geq 0$ be the availability of item type $k \in K$. Now, we consider all solutions $p = (p_1, \dots, p_m) \in \mathbb{R}_+^m$ for which the following three inequalities are satisfied.

$$p(j) \leq b_j, \quad \forall j \in W \tag{5.2}$$

$$p(j) > b_j, \quad \forall j \in J \setminus W \tag{5.3}$$

$$\sum_{j \in W} d_{jk} \leq c_k, \quad \forall k \in K \tag{5.4}$$

Since (5.3) is a strict inequality, we cannot simply find the optimal solution at a vertex of a linear program as we did in the affine pricing problem with unlimited supply. Furthermore, there might exist no envy-free vertex at all. To see this, consider the following example.

Example 5.2.7. *Consider the situation where $J = \{1, 2, 3\}$, $K = \{1, 2\}$ and $c_1 = c_2 = 1$. The valuation constraints are $p(1) = p_1 \leq 1 = b_1$, $p(2) = p_2 \leq 1 = b_2$ and $p(3) = p_1 + p_2 \leq 2 = b_3$. Figure 5.2 displays this setting, where all price vectors $p \in \mathbb{R}_+^m$ in and on the boundary of the gray area are not envy-free. No vertex, represented by the bullets, yields an envy-free solution. However, for some $\varepsilon > 0$, vectors $p = (1, 1 + \varepsilon)$ and $p = (1 + \varepsilon, 1)$ are optimal solutions with revenue 1.*

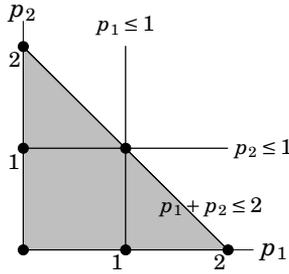


Figure 5.2: Example of an instance of affine pricing with limited supply where there is no envy-free vertex.

Using the insight of the above example, we show how to get a near-optimal envy-free solution by exploring an “ ε -environment” of the vertices as defined in Definition 5.2.2, in case this environment contains envy-free solutions.

Definition 5.2.8 (Near-optimal solution). *A vector $p \in \mathbb{R}_+^m$ is a near-optimal solution if, for a given $\delta > 0$, the revenue obtained by this price vector is at least $(1-\delta)$ times the revenue obtained by any feasible, envy-free vector $\tilde{p} \in \mathbb{R}_+^m$.*

Definition 5.2.9 (Envy-free vertex with limited supply). *A vector $p \in \mathbb{R}_+^m$ is an envy-free vertex of the arrangement of valuation and nonnegativity inequalities if, for some $\varepsilon > 0$, it is characterized by m out of the following $2n+m$ equalities*

$$\begin{aligned} d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m &= b_j, & j \in J \\ d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m &= b_j + \varepsilon, & j \in J \\ p_k &= 0, & k \in K, \end{aligned}$$

such that the coefficient matrix of these m equalities has rank m , and satisfies inequalities (5.2), (5.3) and (5.4).

Given $\delta > 0$, set $\varepsilon = \delta/(nm)$. Now, applying Algorithm 5.1 on the envy-free vertices from Definition 5.2.9, we derive a near-optimal solution to the envy-free affine pricing problem with limited supply. Here, we consider $\binom{2n+m}{m} \in O((n+m)^m)$ systems of m equalities each. In each of these iterations, we solve a linear system in m variables and m constraints to determine the price vector, which takes $O(m^3)$ time. Checking whether all prices are positive takes $O(m)$ time. Computation of the contract prices, winners and the

revenue, and checking whether no item type is oversold takes $O(nm)$ time. We obtain the following corollary.

Corollary 5.2.10. *For a given $\delta > 0$, we can find a near-optimal solution to the envy-free affine pricing problem with limited availability of item types in polynomial time if the number of item types m is constant. The computation time is $O((n+m)^m(m^3+nm)) \in O(n^{m+1})$.*

On the negative side, it turns out that for non-constant m the problem with limited supply seems even harder to approximate than the same problem with unlimited supply.

Theorem 5.2.11. *Consider the affine pricing problem with limited supply and non-constant dimension. For any $\varepsilon > 0$, it is NP-hard to approximate the maximum revenue within a factor $n^{1-\varepsilon}$. This result holds even if all customers have a valuation of one, the availability of each item type is one, and each item type is requested once by at most two customers.*

Proof. We use an approximation preserving reduction from INDEPENDENT-SET. Given a graph $G = (V, E)$, we want to find a maximum cardinality subset $V' \subseteq V$ such that no two vertices in V' are adjacent. Zuckerman (2006) shows that it is NP-hard to approximate the maximum independent set within a factor $|V|^{1-\varepsilon}$, for any $\varepsilon > 0$.

We construct the following instance of the pricing problem. Each vertex $v \in V$ corresponds to a customer and each edge $e \in E$ corresponds to an item type. For every customer v , let $d_{v0} = 1$ and $d_{ve} = 1$ if edge e is incident to vertex v , and 0 otherwise. The valuation b_v is equal to 1. Each item type e is available once. The company determines the prices of the item types $e \in E$, and the price of the contract requested by customer $v \in V$ is equal to $p(v) = d_{v0} + \sum_{e \in E} d_{ve} p_e$.

We claim that an independent set of cardinality s exists in G if and only if there exists a pricing for the above defined instance with total revenue s . Suppose $V' \subseteq V$ is an independent set in G with $|V'| = s$. Then let $p_e = 0$ for all edges $e \in E$ incident to a vertex $v \in V'$, and $p_e > 0$ otherwise. This way the set of winners equals the independent set V' , and therefore no item type is oversold. No customer is envious, as the price of a contract exceeds the valuation for every loser, and we extract a total revenue of s .

Conversely, assume a solution to the pricing problem with total revenue s . Since only one copy of any item type is available, the set of winners must define an independent set in G . As the maximum revenue received from any customer is 1, there exists an independent set of size s in G . \square

5.2.2 Unit-demand customers

So far, we assumed that customers are single-minded, which means that they request one contract and either accept or reject it. However, in this subsection we claim that, for unlimited supply, we can apply the same algorithm for the case with unit-demand customers. That is, each customer can request several different contracts, each with a separate valuation. Every customer accepts at most one of the requested contracts. We assume that the customer selects the contract with the highest *utility*, which is the valuation minus the price, if this utility is nonnegative. For the example in which customers request contracts from a telephone company, we can imagine that customers are interested in different contracts offered by the company, but in the end, they need only one contract to use their telephone.

Let L_j be the set of contracts requested by customer $j \in J$, such that $|L_j| \geq 1$ and $L_j \cap L_k = \emptyset$ for all $j \neq k \in J$, and let $L = \bigcup_{j \in J} L_j$. Notice that if two customers request the same contract, we store both copies in L , for ease of notation. Customer $j \in J$ is interested in at most one contract from subset $L_j \subseteq L$, and has valuation b_ℓ for contract $\ell \in L_j$. Let positive coefficient $d_{\ell k}$ be the demand of customer $j \in J$ for item type $k \in K$ in contract $\ell \in L$. The price for the contract $\ell \in L$ is

$$p(\ell) = d_{\ell 0} + d_{\ell 1}p_1 + \dots + d_{\ell m}p_m.$$

In this section, we redefine a vertex as follows.

Definition 5.2.12 (Vertex with unit-demand customers). *A vector $p \in \mathbb{R}_+^m$ is a vertex of the arrangement of linear inequalities with unit-demand customers if it is characterized by m equalities out of the following $|L| + m$ equalities*

$$\begin{aligned} d_{\ell 0} + d_{\ell 1}p_1 + \dots + d_{\ell m}p_m &= b_\ell, & \ell \in L \\ p_k &= 0, & k \in K, \end{aligned}$$

such that the coefficient matrix of these m equalities has rank m .

As a direct consequence of the definition of a vertex with unit-demand customers and Lemma 5.2.3, we obtain the following corollary.

Corollary 5.2.13. *There exists a vertex as defined in Definition 5.2.12 that is an optimal pricing strategy to the affine pricing problem with unit-demand customers.*

For every vertex with unit-demand customers, we do the following. We determine the price of every contract $\ell \in L$. The *demand set* D_j for customer $j \in J$ is the set of contracts she prefers most, thus $D_j = \{\ell \in L_j : b_\ell - p(\ell) \geq b_{\ell'} - p(\ell'), \forall \ell' \in L_j\}$. We assume that customer $j \in J$ selects a contract $\ell_j \in D_j$ with highest price, as otherwise the company could reduce the prices of this contract with a very small amount, making this the contract with the highest utility for the customer. Then, $W = \{j \in J : p(\ell_j) \leq b_{\ell_j}\}$ and $\Pi = \sum_{j \in W} p(\ell_j)$. Finally, we select the maximum among all values Π , with optimal parameters p_1, \dots, p_m , and set of winners W .

Theorem 5.2.14. *The affine pricing problem with unit-demand customers and unlimited supply can be solved in polynomial time if the number of item types m is constant. The computation time is $O((|L| + m)^m(m^3 + |L|m + |L|n)) \in O(n|L|^{m+1})$.*

Proof. We consider $\binom{|L|+m}{m} \in O((|L| + m)^m)$ systems of m equalities each. In each of these iterations, we solve a linear system in m variables and m constraints, which takes $O(m^3)$ time. It takes $O(m)$ time to verify whether all prices are positive. Computation of the contract prices takes $O(|L|m)$ time, selection of the most preferable contract for every customer takes $O(|L|n)$ time, and computation of the winners and the revenue takes $O(n)$ time. Thus, the time complexity is $O((|L| + m)^m(m^3 + |L|m + |L|n))$, which is in $O(n|L|^{m+1})$ for constant m . \square

Similarly as before, if the number of item types m is not constant, this problem is a generalization of the single item pricing problem with unit-demand customers. As the problem with unit-demand customers is in fact a generalization of the problem with single-minded customers, we have the following corollary.

Corollary 5.2.15. *The affine pricing problem with an unbounded number of item types and unit-demand customers is inapproximable within a semi-logarithmic factor.*

5.3 Local search

The computation time of the affine pricing algorithm (Algorithm 5.1) is exponential in the number of item types. Therefore, we suggest a local search algorithm that finds a near-optimal pricing strategy in polynomial time even if the number of item types m is not constant.

For notational purposes, let $I = \{1, \dots, n, n+1, \dots, n+m\}$ be the *index set of inequalities*. For all $i \in I$, let $p(i) = p(j)$ and $b_i = b_j$ if $i = j \in J$, and $p(i) = -p_k$ and $b_i = 0$ if $i - n = k \in K$. We consider the arrangement of linear inequalities $p(i) \leq b_i$ for all $i \in I$. For this section, we redefine a *vertex*, and also give a definition of an *edge* in the arrangement of linear inequalities.

Definition 5.3.1 (Vertex). *A vertex v is defined by a price vector $p^{(v)} \in \mathbb{R}_+^m$ that is described by a subset $I_v \subset I$ such that $|I_v| = m$ and the coefficient matrix of the equalities $p^{(v)}(i) = b_i$ for all $i \in I_v$ has rank m .*

For every vertex v , we define the contract price for customer $j \in J$ as $p^{(v)}(j) = d_{j0} + d_{j1}p_1^{(v)} + \dots + d_{jm}p_m^{(v)}$. The set of winners is $W^{(v)} = \{j \in J : p^{(v)}(j) \leq b_j\}$. The revenue of vertex v is denoted by $\Pi(v) = \sum_{j \in W^{(v)}} p^{(v)}(j)$.

Definition 5.3.2 (Edge). *An edge $e = \{v, w\}$ between two vertices v and w exists if $|I_v \cap I_w| = m - 1$. Then, we call vertices v and w adjacent, and both are incident to edge e .*

In the local search algorithm, we travel from one vertex to another along the *edges* of the arrangement of linear inequalities. In this sense the algorithm resembles the *simplex algorithm* for linear programming. The simplex algorithm was proposed by George Dantzig² in 1947. This algorithm begins at a vertex of a polytope that represents a feasible region, defined by a series of linear inequalities. Then, it moves along the edges of the polytope until it reaches the vertex of the optimum solution. For further reading, we refer to Dantzig (1963), Papadimitriou and Steiglitz (1998) and Vanderbei (2001).

Although all inequalities are linear, the pricing problem is not a linear program, and therefore we cannot apply the simplex method. The idea of the local search algorithm is to travel along edges of the arrangement of linear inequalities in a steepest ascent fashion. We first give an intuitive description of the algorithm and present the details in Algorithm 5.2. Then, we clarify the procedure in Example 5.3.3.

We start in an arbitrary vertex, say A , described by some set $I_A \subseteq I$ of cardinality m as defined in Definition 5.3.1. We select an arbitrary index $i^A \in I_A$. Then, we explore all vertices v adjacent to A such that $i^A \in I_v$. In other words, we explore all vertices adjacent to A for which equality $p(i^A) = b_{i^A}$ holds. Among these vertices we pick the one that yields the highest

²George Bernard Dantzig (1914-2005) was an American mathematician, and the Professor Emeritus of Transportation Sciences and Professor of Operations Research and of Computer Science at Stanford.

Algorithm 5.2: Local search algorithm

Let $I = \{1, \dots, n, n+1, \dots, n+m\}$ and $\Lambda = \emptyset$. Let $p(i) = p(j)$ and $b_i = b_j$ if $i = j \in J$, and $p(i) = -p_k$ and $b_i = 0$ if $i - n = k \in K$;

Select a vertex v described by set $I_v \subseteq I$ with $|I_v| = m$, and select $i^v \in I_v$;

Let $p(i) = b_i, \forall i \in I_v$ and calculate the price vector $p^{(v)}$, contract prices $p^{(v)}(j)$ for all $j \in J$, the set of winners $W^{(v)}$, and the revenue $\Pi(v)$;

Let $p^* = p^{(v)}, W^* = W^{(v)}$ and $\Pi^* = \Pi(v)$;

while $|I| \geq m$ **do**

$I_w = I_v$;

foreach set $I' \subset (I_v \setminus i^v)$ with $|I'| = m - 2$ **do**

foreach $i \in I \setminus I_v$ **do**

$I_u = i^v \cup I' \cup i$;

 Let $p(i') = b_{i'}, \forall i' \in I_u$ and determine $p^{(u)}, W^{(u)}$ and $\Pi(u)$;

if $\Pi(u) > \Pi^*$ **then**

$p^* = p^{(u)}, W^* = W^{(u)}, \Pi^* = \Pi(u)$ and $I_w = I_u$;

end

end

end

$\Lambda = \Lambda \cup \{i^v\}$ and $i^w = I_w \setminus I_v$;

if $i^w \in I$ **then**

$I_v = I_w$ and $i^v = i^w$;

foreach $i \in \Lambda$ **do**

if $i \notin I_v$ **then**

$I = I \setminus i$ and $\Lambda = \Lambda \setminus i$;

end

end

else

$I = I \setminus \Lambda, \Lambda = \emptyset$;

if $|I| \geq m$ **then**

 Select some set $I_v \subseteq I$ and pick $i^v \in I_v$;

 Determine $p^{(v)}, W^{(v)}$ and $\Pi(v)$;

if $\Pi(v) > \Pi^*$ **then**

$p^* = p^{(v)}, W^* = W^{(v)}$ and $\Pi^* = \Pi(v)$;

end

end

end

end

revenue, that is, we locally optimize the revenue. If this revenue is higher than the revenue generated in vertex A , we accept this vertex as the new solution, say B , and let $i^B = I_B \setminus I_A$. Now, we find the vertex with highest revenue adjacent to B that satisfies $p(i^B) = b_{i^B}$ but is not incident to edge $\{A, B\}$. If we arrive at a vertex for which no adjacent vertex yields a higher revenue, we stop the procedure. However, if there are at least m equalities left that we did not explore before, characterized by indices in set $I' \subseteq I$ with $|I'| \geq m$, we restart the algorithm at an arbitrary vertex defined by m equalities for which the indices are chosen from set I' .

Example 5.3.3. Consider the instance with four customers and three item types, as displayed in Figure 5.3. The customers' valuation inequalities are given by $p(1) = \frac{3}{2}p_1 + \frac{3}{2}p_2 + p_3 \leq 36 = b_1$, $p(2) = 4p_1 + p_2 + p_3 \leq 32 = b_2$, $p(3) = \frac{3}{2}p_2 + p_3 \leq 24 = b_3$, and $p(4) = \frac{1}{2}p_1 + \frac{2}{5}p_2 + p_3 \leq 16 = b_4$. Moreover, $p(5) = -p_1 \leq 0 = b_5$, $p(6) = -p_2 \leq 0 = b_6$ and $p(7) = -p_3 \leq 0 = b_7$, and therefore $I = \{1, \dots, 7\}$.

We start the procedure in vertex A with $p^{(A)} = (0, 0, 36)$ described by $I_A = \{1, 5, 6\}$ and we select element $i^A = 1 \in I_A$. The revenue is $\Pi(A) = 36$. We explore all vertices v adjacent to A satisfying constraint $p(1) = b_1$. In the figure, this is displayed by the arrows on edges leaving vertex A . Vertex B described by $I_B = \{1, 4, 5\}$ with $p^{(B)} = (0, 18\frac{2}{11}, 8\frac{8}{11})$ has the highest revenue of $\Pi(B) = 78\frac{10}{11}$ among all adjacent vertices, and is thus the local optimum if we start in A . We travel from A to B , and update $\Lambda = \{1\}$ and $i^B = I_B \setminus I_A = 4$. As all customers in Λ are also in I_B , we keep set I as it is. Now, we find all vertices v adjacent to B with $p(4) = b_4$, thus explore all edges incident to B except the edge we came from. Again, this is displayed by arrows on edges leaving vertex B . Vertex C with $p^{(C)} = (0, 7\frac{3}{11}, 13\frac{1}{11})$ has the highest revenue of $\Pi(C) = 84\frac{4}{11}$, and as $\Pi(C) > \Pi(B)$, we travel to vertex C with $I_C = \{3, 4, 5\}$. We update $\Lambda = \{1, 4\}$ and $i^C = I_C \setminus I_B = 3$. However, as $1 \notin I_C$ we have $I = \{2, \dots, 7\}$ and $\Lambda = \{4\}$. Similarly, we travel from C to D with $p^{(D)} = (3\frac{7}{83}, 8\frac{56}{83}, 10\frac{82}{83})$, $I_D = \{2, 3, 4\}$ and $i^D = 2$, and update Λ to $\{2, 3, 4\}$ whereas I does not change. There is no vertex adjacent to D that has a higher revenue than $\Pi(D) = 100\frac{52}{83}$. Therefore, we remove Λ from I which leaves us with $I = \{5, 6, 7\}$, and empty set Λ . As $|I| = m$, we explore vertex E with $I_E = \{5, 6, 7\}$. This vertex nor its adjacent vertices have a higher revenue than $\Pi(D)$, thus the algorithm stops and outputs the prices $p^{(D)}$.

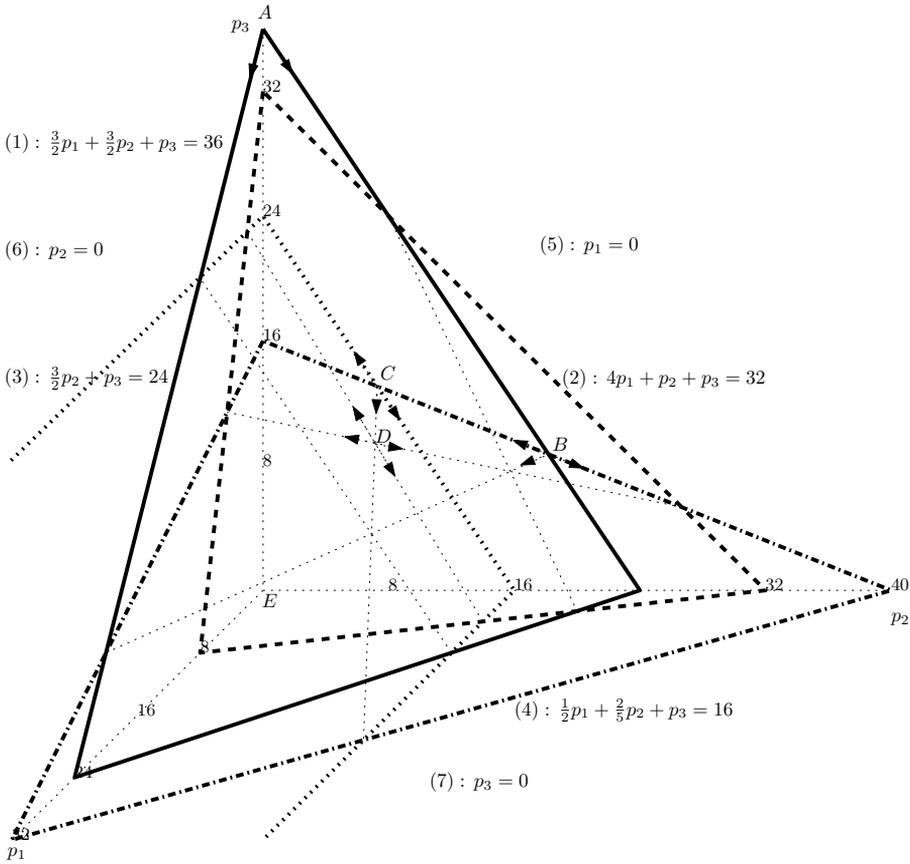


Figure 5.3: Graphical representation of the local search algorithm on an instance with $m = 3$.

Theorem 5.3.4. *Algorithm 5.2 runs in $O(nm^5 + n^2m^4 + n^3m^2)$ time, which is in $O(n^3)$ if the number of item types is constant.*

Proof. Every index $i \in I$ is selected at most once to be i^v , as after the selection it is added to Λ ; that is $O(n + m)$. Then, we select every subset I' from $I_v \setminus i^v$ of size $m - 2$ once. As $|I_v \setminus i^v| = m - 1$, there are exactly $\binom{m-1}{m-2} = m - 1$ different subsets I' possible. Moreover, $|I \setminus I_v| \leq n$. In each iteration, we solve a linear system of m constraints and m variables to find price vector p , which takes $O(m^3)$ time. Checking if all prices are positive takes $O(m)$ time. Computation of contract prices, winners and the revenue takes $O(nm)$. Then, we update i^v , Λ and I . If the new i^v is an element of I then we check at most

$|\Lambda| \leq m$ times whether the equality with index i^v in Λ partly defines the current vertex. Otherwise, we select a new vertex and determine its price vector in $O(m^3)$ time, and revenue and winners in $O(nm)$ time. Thus, the total running time is $O(nm^5 + n^2m^4 + n^3m^2)$, which is in $O(n^3)$ if the number of item types m is constant. \square

5.4 Computational study

We apply the affine pricing algorithm and the local search algorithm to determine a pricing strategy for a telephone operator. The data we use resemble reality. The item types are available in unlimited supply, as we are selling digital goods. We assume customers to be single-minded, that is, each customer either accepts the offer for the contract or she leaves to a competitor. In this practical application, we determine the start-up tariff, a price per minute for calling within the country and abroad, and a price per sent text message.

The data we use contain detailed information about the phone usage of many customers. For each customer, we can exactly determine the demand for the different item types and therefore the contract she requests. Let p_1 be the start-up price for a call, p_2 the price per minute for calling domestically, p_3 the price per text message, and p_4 the price per minute for calling abroad. Then, the price for the contract of customer $j \in J$ is defined as

$$p(j) = d_{j1}p_1 + d_{j2}p_2 + d_{j3}p_3 + d_{j4}p_4,$$

where d_{j1} is the number of calls customer j wants to make, d_{j2} is the number of minutes she calls within the country, d_{j3} is the number of text messages she sends, and d_{j4} is the number of minutes she calls abroad.

We assume that customers are rational, and therefore select the cheapest offer for their contract. Thus, a customer only accepts our offer for her contract if we offer the cheapest price in the market. Therefore, we calculate the *valuation* of a customer as follows. First, we consider all existing pricing strategies of the company we observe, and of some large competing telephone companies. Then, we calculate for every customer what the price of her contract would be for each of the current pricing strategies. The cheapest of these prices will be the valuation. However, any other valuation determined by market research can be used as well. The goal of this practical research is: Given that the competing operators keep their prices fixed, find the pricing strategy that maximizes the revenue.

5.4.1 Results

We conduct several tests on the data we received on a computer with an Intel Core2Duo CPU running at 2.40GHz and 3.25 GB of Ram. Besides optimizing all four item type prices as described above ($m = 4$), we also optimize prices while having the start tariff of a call for free ($m = 3$), thus $p(j) = d_{j2}p_2 + d_{j3}p_3 + d_{j4}p_4$, and moreover, keeping the price for calling abroad at the original tariff of the telephone company of 0.75 euro ($m = 2$), that is $p(j) = d_{j2}p_2 + d_{j3}p_3 + 0.75d_{j4}$. For these three tests with an extensive data set containing data of 6160 customers, the local search algorithm gives the *optimal* solution. The running time of this algorithm is significantly smaller than the running time of the affine pricing algorithm, as shown in Table 5.1. For the case with $m = 4$ the exact running time of the affine pricing algorithm is not presented, as the running time is extremely large which is not practically useful.

	$m = 2$	$m = 3$	$m = 4$
Current revenue	21940.97	25888.40	25888.40
Optimal revenue	33515.80	38648.50	44917.60
Runtime affine pricing	1554 sec.	975453 sec.	> 4 months
Runtime local search	2 sec.	2 sec.	13 sec.

Table 5.1: Results in euro for instance with 6160 customers in dimension m .

In the above test on the complete data set, the local search algorithm performed perfectly as it always resulted in the optimal solution. Unfortunately, there is no guarantee that the local search algorithm will produce the optimum. However, Algorithm 5.2 is a practically powerful tool as its running time is significantly smaller than the running time of the exact algorithm, and by construction, will produce a fairly good solution. To stress this, we perform nine tests on several random samples of the data set with $m = 4$. We use samples of 100, 200, and 300 customers. The results of the affine pricing algorithm (A), the local search algorithm (L) and the current revenue (C) are presented in Table 5.2, where the runtime is in seconds. Figure 5.4 presents the relative deviation from the optimal revenue.

These results indicate that the revenue obtained by the local search algorithm is not always optimal, but it is not far from it either. The running time of the local search algorithm is a lot shorter than the running time of the affine pricing algorithm, especially for large samples.

Test1 - $n = 100$			Test2 - $n = 100$			Test3 - $n = 100$		
Method	Revenue	Sec.	Method	Revenue	Sec.	Method	Revenue	Sec.
A	11945.10	22	A	8365.76	22	A	173.22	21
L	11945.10	0	L	7906.89	0	L	172.91	0
C	4905.02		C	5316.53		C	136.96	
Test4 - $n = 200$			Test5 - $n = 200$			Test6 - $n = 200$		
Method	Revenue	Sec.	Method	Revenue	Sec.	Method	Revenue	Sec.
A	3405.37	416	A	8018.33	415	A	3530.61	414
L	3405.37	2	L	7853.29	0	L	3482.65	0
C	2263.08		C	4162.48		C	2337.43	
Test7 - $n = 300$			Test8 - $n = 300$			Test9 - $n = 300$		
Method	Revenue	Sec.	Method	Revenue	Sec.	Method	Revenue	Sec.
A	5054.56	2467	A	6448.82	2432	A	5681.08	2450
L	4934.63	1	L	6436.11	0	L	5681.08	0
C	2636.76		C	4015.67		C	3691.81	

Table 5.2: Results for samples compared to the optimal revenue.

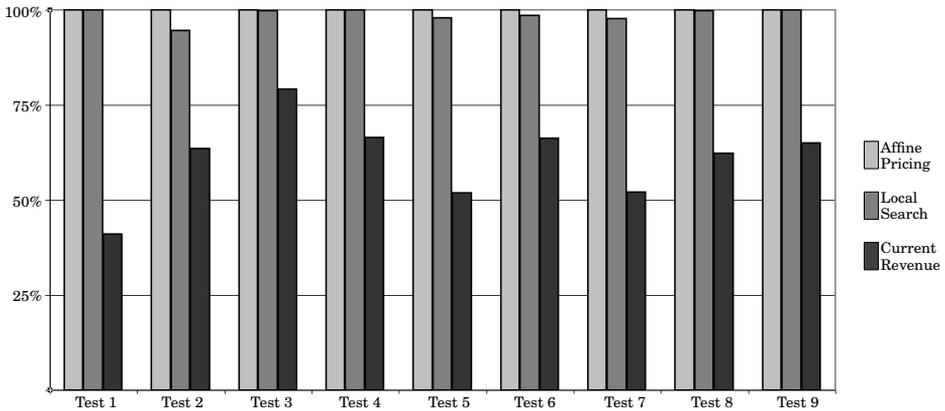


Figure 5.4: Relative deviation from the optimal solution of affine pricing, local search and current solution from the optimum.

5.5 Conclusion

Affine pricing can be applied in many different types of companies because of its very general structure. Also, as long as the number of different item types is constant, we can solve the pricing problem with unlimited supply in polynomial time, irrespective of the profile of the customers (single-minded or unit-demand), and we derive a near-optimal solution if the supply is lim-

ited. However, if the number of item types is not constant, the hardness results from single item pricing apply to affine pricing as well. Moreover, the envy-free problem with limited supply is even harder to approximate than the problem with unlimited supply.

Challenged by the large amount of data received from a telephone operator, we have the opportunity to practically test our algorithms. In the computational study, the customers are single-minded and the supply is unlimited. The theoretical computation time for the algorithm is $O(n^{m+1})$ for a constant number of item types m . Although the affine pricing algorithm always finds the optimal pricing strategy for the given instances, the running time appeared to be too high for practical purposes. Therefore, we introduce a local search algorithm that runs in $O(n^3)$ time for constant number of item types m . We were not able to prove any bound on the performance of the algorithm, but on the complete and sampled instances, the results are very promising and the running time is very low in practice.

Chapter 6

Price Strategy Implementation

Consider a situation in which a company sells several different item types to a set of customers. Every customer requests a contract, based on the types of items for sale and her personal demand. Currently, the company is not satisfied with the pricing strategy and wants to implement new prices. Implementing these new prices in one single time step might not be desirable, for example, because of the change in contract prices for the customers. Therefore, the company changes the prices gradually, such that the prices charged to a subset of the customers, the target market, do not differ too much from one period to the next. We propose a polynomial time algorithm to implement the new prices in the minimum number of time periods needed, given that the prices charged to the customers in the target market increase by at most a factor $1 + \delta$, for some $\delta > 0$. Furthermore, we address the problem to maximize the revenue when a maximum number of time periods is predetermined. For this problem with integer price vectors, we describe a dynamic program. We extend this result and present an approximation algorithm for the problem with continuous prices. Moreover, we propose a local search algorithm. Also, we present the integer program that models this problem and conduct a practical study.¹

¹This chapter is based on Berger, Grigoriev, and van Loon (2008).

6.1 Introduction

A company owns a set of different item types for sale, $K = \{1, \dots, m\}$. We assume that every item type is available in unlimited supply. The set of potential customers is denoted by $J = \{1, \dots, n\}$. Every customer $j \in J$ has a personal demand $d_{jk} \geq 0$ for every item type $k \in K$. The combination of the demand for all item types is called a customer's *contract*. Every customer $j \in J$ is single-minded and has a valuation b_j for the contract, which is the maximum amount she is willing to pay. Let $B = \max_{j \in J} \{b_j\}$ be an upper bound on the valuations.

In this chapter, we regard the problem that the company faces when implementing a new price vector $p^* \in \mathbb{R}_+^m$. This price vector might be the optimal price vector obtained by solving the affine pricing algorithm as discussed in Chapter 5, but can also be any other vector that the company wishes to implement. We assume that the current price vector is $p^0 \neq p^*$. Immediate introduction of the new prices in the next period (e.g. week, month) might not be desirable. For example, a sudden huge difference in prices may change the perception of customers or the position in the market in comparison to the competing companies. Therefore, the company wants to change prices *gradually*. Let $T = \{0, \dots, t_{\max}\}$ be the set of time periods, where $t = 0$ is the current time and $t = t_{\max}$ is the end time of the implementation. For notational purposes, let $T^+ = T \setminus \{0\}$.

For every time step $t \in T$, the company has to set price vector $p^t = (p_1^t, \dots, p_m^t) \in \mathbb{R}_+^m$. Given these prices, the contract price for customer $j \in J$ is an affine function on her personal demand

$$p^t(j) = d_{j0} + d_{j1}p_1^t + \dots + d_{jm}p_m^t.$$

We call a customer a winner if she can afford her contract and receives it. We denote the set of winners by $W^t = \{j \in J : p^t(j) \leq b_j\}$. The company's revenue in time period t is $\Pi^t = \sum_{j \in W^t} p^t(j)$. The company's total revenue is equal to

$$\Pi = \sum_{t \in T} \Pi^t = \sum_{t \in T} \sum_{j \in W^t} p^t(j).$$

Given price vector $p^* \in \mathbb{R}_+^m$, we can easily determine for which customers the price of the contract does not exceed valuation, that is, $W^* = \{j \in J : p^*(j) \leq b_j\}$. We call W^* the *target market*. When implementing the optimal pricing strategy, we have to satisfy the restriction that for these customers the price of the contract may increase at most a factor of $(1 + \delta)$ between

two consecutive periods for some positive δ , which will be predetermined by the company. That is, given price vectors p^0 , p^* and δ , the company has to satisfy the following two constraints.

Constraint 6.1. $p^t(j) \leq (1 + \delta)p^{t-1}(j), \forall j \in W^*, t \in T^+$.

Constraint 6.2. $p_k^{t_{\max}} = p_k^*, \forall k \in K$.

We address two different problems in this chapter, which are formally defined below.

Problem 6.1. *Given p^0 , p^* and $\delta > 0$, find price vectors p^t for all $t \in T^+$, so as to minimize t_{\max} subject to Constraints 6.1 and 6.2.*

Problem 6.2. *Given p^0 , p^* , $\delta > 0$ and t_{\max} that is large enough to satisfy Constraints 6.1 and 6.2, find price vectors p^t for all $t \in T^+$ subject to Constraints 6.1 and 6.2 so as to maximize $\sum_{t \in T} \Pi^t$.*

For Problem 6.1, we show how to calculate the minimum number of time periods and we present an algorithm to find the price vectors with computation time $O(nm + m \log B)$. For Problem 6.2 with integer price vectors, we describe a dynamic program that runs in pseudo-polynomial time if the number of item types m is constant. For the problem with continuous price vectors, the dynamic program gives an approximate solution with performance guarantee $(1 + \varepsilon)$ for any $\varepsilon > 0$, and a computation time of $O(t_{\max}^2 (B/\varepsilon)^{2m} nm)$. Then, we introduce a local search algorithm that, in every time period $t > 0$, selects a pricing strategy that yields the highest total revenue to the company, but also satisfies the above constraints. Although we were not able to estimate the performance guarantee on the solution of this algorithm, computational studies show that the obtained revenue is not far from the optimal total revenue calculated by solving an integer linear program.

6.2 Implementation in minimal time

We assume that the price of any item type in the current solution is at least equal to some very small $\varepsilon > 0$, that is, $p_k^0 \geq \varepsilon$ for all $k \in K$. Let q_j denote the ratio between customer j 's contract price in the optimal and current pricing strategy and let q be the maximum ratio among all customers in the target market, that is,

$$q = \max_{j \in W^*} \{q_j\} = \max_{j \in W^*} \left\{ \frac{p^*(j)}{p^0(j)} \right\}.$$

Theorem 6.2.1. *Given current price vector p^0 , the minimum number of steps to implement price vector p^* is $t_{\max} = t_q$, where $t_q = \max\{1, \lceil \log_{1+\delta} q \rceil\}$.*

Proof. Since $p^0 \neq p^*$, $t_{\max} \geq 1$, that is, we need at least one step to implement the new prices. By Constraint 6.1 we know that

$$p^*(j) \leq (1 + \delta)^{t_{\max}} p^0(j), \quad \forall j \in W^*.$$

By definition of q , there exists some customer $j' \in W^*$ such that

$$q = \frac{p^*(j')}{p^0(j')} \leq (1 + \delta)^{t_{\max}} \Leftrightarrow t_{\max} \geq \log_{1+\delta} q. \quad (6.1)$$

Combining inequality (6.1) with the facts that the minimum number of time periods needed is integer and at least equal to one, we have that $t_{\max} \geq t_q = \max\{1, \lceil \log_{1+\delta} q \rceil\}$.

We construct a sequence of price vectors located on the straight line in \mathbb{R}_+^m between p^0 and p^* . For every such price vector p^t , $t \in T$, we know that

$$p^t = \mu^t p^* + (1 - \mu^t) p^{t-1}, \quad (6.2)$$

for some $\mu^t \in [0, 1]$. For every customer $j \in J$ we have

$$\begin{aligned} p^t(j) &= d_{j0} + d_{j1}((p_1^* - p_1^{t-1})\mu^t + p_1^{t-1}) + \dots + d_{jm}((p_m^* - p_m^{t-1})\mu^t + p_m^{t-1}) \\ &= p^{t-1}(j) + (p^*(j) - p^{t-1}(j))\mu^t. \end{aligned} \quad (6.3)$$

Now, for every $t \in T^+$, we define a set $W_t^* = \{j \in W^* : p^*(j) > p^{t-1}(j)\}$ and

$$\mu^t = \min \left\{ \min_{j \in W_t^*} \left\{ \frac{\delta p^{t-1}(j)}{p^*(j) - p^{t-1}(j)} \right\}, 1 \right\}. \quad (6.4)$$

Notice that if $W_t^* = \emptyset$ then $\mu^t = 1$ and we are able to reach solution p^* at time t , as

$$p^t(j) = p^{t-1}(j) + (p^*(j) - p^{t-1}(j)) = p^*(j) \leq p^{t-1}(j), \quad \forall j \in W^*.$$

Otherwise, combining Equations (6.3) and (6.4), we have for every $t \in T^+$ and $j \in W_t^*$ that

$$p^t(j) \leq p^{t-1}(j) + (p^*(j) - p^{t-1}(j)) \frac{\delta p^{t-1}(j)}{p^*(j) - p^{t-1}(j)} = (1 + \delta) p^{t-1}(j),$$

and for all $j \in W^* \setminus W_t^*$ we have that

$$p^t(j) \leq p^{t-1}(j) < (1 + \delta)p^{t-1}(j),$$

thus Condition 6.1 is satisfied for all $j \in W^*$.

We know that

$$\arg \min_{j \in W_t^*} \left\{ \frac{\delta p^{t-1}(j)}{p^*(j) - p^{t-1}(j)} \right\} = \arg \max_{j \in W_t^*} \left\{ \frac{p^*(j)}{p^{t-1}(j)} \right\} = \arg \max_{j \in W^*} \left\{ \frac{p^*(j)}{p^{t-1}(j)} \right\},$$

where the last equality holds by definition of W_t^* . We claim that, for every $t \in T^+$, the customer that minimizes μ^t also maximizes q , that is,

$$\arg \max_{j \in W_t^*} \left\{ \frac{p^*(j)}{p^{t-1}(j)} \right\} = \arg \max_{j \in W^*} \left\{ \frac{p^*(j)}{p^0(j)} \right\}.$$

This claim is trivially true for $t = 1$, again by definition of W_t^* . Assume the claim is true for all time periods $1, \dots, t-1$. Then, for period t we have

$$\begin{aligned} \arg \max_{j \in W_t^*} \left\{ \frac{p^*(j)}{p^{t-1}(j)} \right\} &= \arg \max_{j \in W_t^*} \left\{ \frac{p^*(j)}{\mu^{t-1} p^*(j) + (1 - \mu^{t-1}) p^{t-2}(j)} \right\} \\ &= \arg \max_{j \in W_t^*} \left\{ \frac{p^*(j)}{p^{t-2}(j)} \right\} = \arg \max_{j \in W^*} \left\{ \frac{p^*(j)}{p^0(j)} \right\}, \end{aligned}$$

where the last equality holds due to the induction hypothesis. Thus, the claim holds for all $t \in T^+$.

Consequently, at step $t_q = \max\{1, \lceil \log_{1+\delta} q \rceil\}$ we reach p^* . Thus, $t_{\max} \leq t_q$. As we also showed that $t_{\max} \geq t_q$, the claim is proven. \square

Using the steps in the proof above, we define the *straight line algorithm* as follows. Given $\delta > 0$ and price vectors p^0 and p^* , let $j' \in W^*$ be the customer with $q = p^*(j')/p^0(j')$. Then, we have

$$\mu^t = \min \left\{ \frac{\delta p^{t-1}(j')}{p^*(j') - p^{t-1}(j')}, 1 \right\},$$

for all time steps $t = 1, \dots, t_q$, where $t_q = \max\{1, \lceil \log_{1+\delta} q \rceil\}$. Let $p_k^t = (p_k^* - p_k^{t-1})\mu^t + p_k^{t-1}$ be the price of item type $k \in K$ in time period $t = 1, \dots, t_q - 1$. Finally, in time period t_q , we implement price vector p^* .

Theorem 6.2.2. *The straight line algorithm finds a price implementation from p^0 to p^* in $O(nm + m \log B)$ time.*

Proof. The price implementation generated by the straight line algorithm satisfies Condition 6.1 as shown in the proof of Theorem 6.2.1 and Condition 6.2 by definition of the final step in the algorithm. Also, it uses t_q steps to implement the new price vector, which is minimal. Thus, the algorithm solves Problem 6.1. Finding customer $j' \in W^*$ who maximizes q takes $O(nm)$ time, as we calculate the ratio between the contract prices for the new and current price vectors, for every customer in the target market. For all $t = 1, \dots, t_q - 1$, where $t_q = \max\{1, \lceil \log_{1+\delta} q \rceil\}$, we need to find μ^t which can be done in $O(m)$ time, as we need to calculate $p^{t-1}(j')$. Then, implementing the price vector p^t takes $O(m)$ time, which results in a total running time of $O(nm + t_q m)$. Integer t_q is bounded from above by the logarithm of the largest valuation, as $q = \max_{j \in W^*} \{p^*(j)/p^0(j)\}$ only depends on the contract prices of the *winners* given price vector p^* . Consequently, the computation time of the straight line algorithm is $O(nm + m \log B)$. \square

6.3 Maximize total revenue

Let t_{\max} be the number of time periods the company wants to use to implement pricing strategy p^* , and let $T = \{0, 1, \dots, t_{\max}\}$. In Problem 6.2 we want to find the strategy that maximizes the total revenue over t_{\max} periods, that is, $\Pi = \sum_{t \in T} \Pi^t = \sum_{t \in T} \sum_{j \in W^t} p^t(j)$. Notice that W^t , $t \in T$, is a subset of J , not just of the target market W^* .

In this section, we first address Problem 6.2 restricted to integer prices. We present a dynamic programming algorithm that solves this problem in pseudo-polynomial time if the number of item types m is constant. Then, using a straightforward scaling argument, we transform the algorithm to a $(1+\varepsilon)$ -approximation algorithm for the same problem with continuous prices, for any desired precision $\varepsilon > 0$. Finally, we present a local search algorithm.

6.3.1 Dynamic program

Given $\delta > 0$ and *integer* price vectors p^0 and p^* , we find the integer price vectors to optimize the total revenue by a dynamic program that runs in pseudo-polynomial time if the number of item types m is constant. This dynamic program determines the longest path from source u to sink v in an acyclic digraph $D = (N, A)$. We construct digraph D consisting of $t_{\max} + 1$ layers as follows. A node is of the form (t, p^t) , where t is the time period, or layer, and p^t is an integer price vector. Arcs are only created between two

consecutive layers. In layer 0, there is only node $u = (0, p^0)$, and in layer t_{\max} there is only node $v = (t_{\max}, p^{t_{\max}})$, where $p^{t_{\max}} = p^*$. There exists an arc between node $(t-1, p^{t-1})$ and node (t, p^t) if

- $t \in T^+$,
- p^{t-1} and p^t are integer price vectors,
- $p^t(j) \leq (1 + \delta)p^{t-1}(j)$ for all $j \in W^*$.

The length of the arc is Π^t . The longest $u - v$ path corresponds to the pricing strategy that yields maximum total revenue. Note that the length of this path, plus Π^0 , is equal to the total revenue.

Theorem 6.3.1. *The dynamic programming algorithm solves Problem 6.2 with integer price vectors in time $O(t_{\max}^2 B^{2m} nm)$.*

Proof. Correctness of the dynamic program follows from the fact that we enumerate over all possible integer price vectors. The number of nodes in layer t is $O(B^m)$, which means that in total we have $O(t_{\max} B^m)$ nodes in the digraph. The number of arcs is $O(t_{\max}^2 B^{2m})$. Determining the length of an arc, Π^t , takes $O(nm)$ time. Finding the longest $u - v$ path in a digraph is linear in the number of arcs (Ahuja, Magnanti, and Orlin 1993). So the claimed complexity follows. \square

In order to solve the problem with continuous price vectors, we allow prices that are integer multipliers of ε , for some $\varepsilon > 0$. We derive the following corollary.

Corollary 6.3.2. *The dynamic programming algorithm is a $(1 + \varepsilon)$ -approximation algorithm running in $O(t_{\max}^2 (B/\varepsilon)^{2m} nm)$ time, for any $\varepsilon > 0$.*

6.3.2 Local search

In this section, we assume that price vector p^* is the optimal price vector for the given instance of the affine pricing problem. The idea of the local search algorithm is to implement the optimal price vector in every time period such that for every customer the contract price is at most equal to the minimum of the valuation and $(1 + \delta)$ times the contract price in the previous time period. To find this optimal price vector, we execute the affine pricing algorithm (Algorithm 5.1) in every time period $t \in T^+$. A formal description is presented in Algorithm 6.1, which is further clarified in Example 6.3.3.

Algorithm 6.1: Local search algorithm

Given p^0 , p^* and $\delta > 0$. Let $\Pi^* = \Pi^0$;

foreach $t \in T^+$ **do**

Let $\Pi^t = 0$, $p^t = 0$ and $W^t = J$;

foreach set of m equalities out of the $|W^*| + m$ equalities
 $p^t(j) = \min\{b_j, (1 + \delta)p^{t-1}(j)\}$, $j \in W^*$, and $p_k^t = 0$, $k \in K$, such that
the coefficient matrix of these m equalities has rank m **do**

Determine price vector p that is characterized by these m
equalities;

if $p_k \geq 0$ for all $k \in K$ **then**

Let $p(j) = d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m$ be the price of the
contract requested by customer $j \in J$;

Let $W = \{j \in J : p(j) \leq b_j\}$ be the set of winners;

Let $\Pi = \sum_{j \in W} p(j)$ be the total revenue;

if $\Pi > \Pi^t$ **then**

| Let $\Pi^t = \Pi$, $p^t = p$ and $W^t = W$;

end

end

end

$\Pi^* += \Pi^t$;

end

Note that this algorithm assures that as soon as we are able to reach the optimal price vector p^* , we will stay at this level as there is obviously no better price vector in reach.

Example 6.3.3. Consider a setting with three customers and two item types, and $\delta = 1$. The valuation constraints $p(j) \leq b_j$ are $p(1) = 16p_1 + 32p_2 \leq 512 = b_1$, $p(2) = 20p_1 + 20p_2 \leq 400 = b_2$ and $p(3) = 28p_1 + 16p_2 \leq 448 = b_3$, displayed by the solid lines in Figure 6.1. Using the affine pricing algorithm, we know that the optimal pricing is $p^* = (8, 12)$. The current price vector is $p^0 = (2, 1)$. With this current price vector, customer 1 pays 64, customer 2 pays 60 and customer 3 pays 72. We create an arrangement of linear inequalities

$$\begin{array}{rcl}
 16 & p_1^1 & + \quad 32 & p_2^1 & \leq \min\{512, (1 + \delta)64\} = 128 \\
 20 & p_1^1 & + \quad 20 & p_2^1 & \leq \min\{400, (1 + \delta)60\} = 120 \\
 28 & p_1^1 & + \quad 16 & p_2^1 & \leq \min\{448, (1 + \delta)72\} = 144 \\
 & p_1^1 & & & \geq 0 \\
 & & & p_2^1 & \geq 0.
 \end{array}$$

The first three constraints are represented by the dotted lines in the figure. The optimal price vector is $p^1 = (4, 2)$. Repeating this procedure, in which the constraints are represented by the dashed lines, leads to $p^2 = (8, 4)$. Then, we get $16p_1^2 + 32p_2^2 \leq \min\{512, (1 + \delta)256\} = 512$, $20p_1^2 + 20p_2^2 \leq \min\{400, (1 + \delta)240\} = 400$, and $28p_1^2 + 16p_2^2 \leq \min\{448, (1 + \delta)288\} = 448$. Thus, this gives the arrangement defined by the valuation constraints. The optimal solution p^* is reachable, so the algorithm selects this price vector and keeps selecting it for all periods $t \geq 3$, $t \in T$.

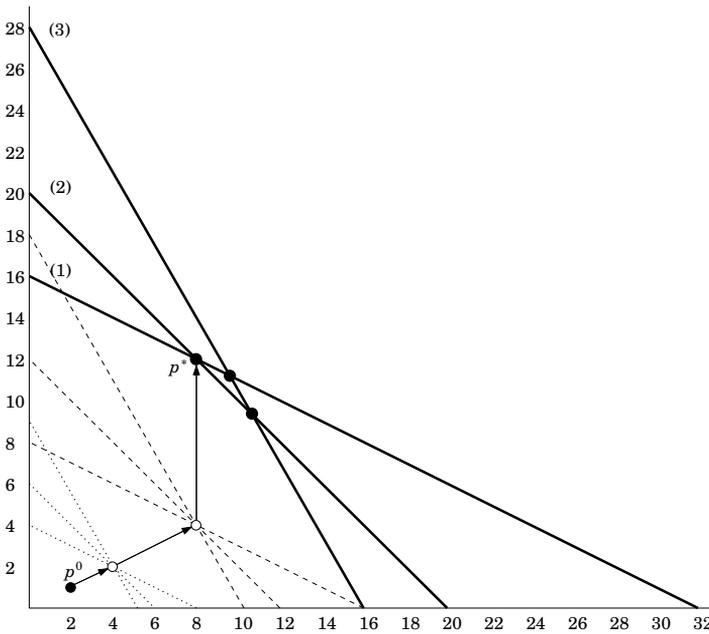


Figure 6.1: Graphical representation of an instance for $m = 2$.

Theorem 6.3.4. Local search algorithm 6.1 runs in $O(t_{\max}(n + m)^m(m^3 + nm))$ time, which is in $O(t_{\max}n^{m+1})$ if the number of item types m is constant.

Proof. In the affine pricing algorithm that is applied during the local search, we consider $\binom{|W^*|+m}{m} \in O((n + m)^m)$ systems of m equalities each. In each of these iterations, we solve a linear system in m variables and m constraints to determine the price vector, which takes $O(m^3)$ time. Checking if all prices are positive takes $O(m)$ time and the computation of the contract prices, winners, and the revenue takes $O(nm)$ time. We perform these steps for all $t \in T^+$, so the claimed complexity follows. \square

6.4 Computational results

We apply the *straight line algorithm* and the *local search algorithm* to implement the optimal pricing strategy for a telephone operator. The item types are available in unlimited supply, as we are selling digital goods. We assume customers to be single-minded, that is, each customer either accepts the offer for the contract or she leaves to a competitor. In this practical application, we determine the start-up price for a call, p_1 , the price per minute for calling domestically, p_2 , the price per text message, p_3 , and the price per minute for calling abroad, p_4 . The data we use contain detailed information about the phone usage of many customers, thus we are able to determine the contract requested by each customer. The price for the contract of customer $j \in J$ is defined as $p(j) = d_{j1}p_1 + d_{j2}p_2 + d_{j3}p_3 + d_{j4}p_4$, where d_{j1} is the number of calls customer j wants to make, d_{j2} is the number of minutes she calls within the country, d_{j3} is the number of text messages she sends, and d_{j4} is the number of minutes she calls abroad. Notice that $d_{j0} = 0$. We assume that customers are rational, and therefore select the cheapest offer for their contract. Thus, a customer only accepts our offer for her contract if we offer the cheapest price in the market. Therefore, we define the valuation of a customer as the cheapest price for her contract at any competitor. We apply the described algorithms to three different samples of the data set. Every sample contains ten customers. The current price vector in euro is $p^0 = (0.01, 0.06, 0.01, 0.75)$ and $\delta = 0.05$.

In this section we determine the minimum number of time periods, and the number of periods used by the local search algorithm to implement the optimal prices. Then, we set $t_{\max} = 40$ and find the optimal pricing strategy. This strategy is calculated by the integer program described below.

$$\begin{array}{ll}
 \max & \sum_{t=0}^T \sum_{j \in J} (\pi_j^t - x_j^t b_j) \\
 \text{s.t.} & p^t(j) \leq (1 + \delta)p^{t-1}(j) \quad \forall t \in T^+, \forall j \in W^* \\
 & \pi_j^t \leq b_j \quad \forall t \in T, \forall j \in J \\
 & \pi_j^t \leq p^t(j) \quad \forall t \in T, \forall j \in J \\
 & p^t(j) \leq (1 + \delta)Bx_j^t + b_j \quad \forall t \in T, \forall j \in J \\
 & p_k^{t_{\max}} = p_k^* \quad \forall k \in K \\
 & p_k^t \geq 0 \quad \forall t \in T, \forall k \in K \\
 & \pi_j^t \geq 0 \quad \forall t \in T, \forall j \in J \\
 & x_j^t \in \{0, 1\} \quad \forall t \in T, \forall j \in J
 \end{array}$$

In an optimal solution at time $t \in T$, we know that variable $\pi_j^t = p^t(j)$ if $j \in W^t$ and $\pi_j^t = b_j$ otherwise. Binary variable x_j^t is equal to 0 if $j \in W^t$ and 1 otherwise.

Sample 1 In the first sample, the optimal price vector is $p^* = (0.175136, 0.170037, 0.087114, 0.766734)$. The minimum number of time steps needed to implement p^* starting from p^0 is 31. The total revenue generated by the straight line algorithm is 64230.64. The optimal revenue for implementing p^* in 31 steps is 70936.90, which means that the revenue of the straight line algorithm is 90.55% of the optimal revenue. The local search algorithm needs at least 34 steps to implement p^* . However, running this algorithm for 31 steps generates a total revenue of 70968.47, which is even larger than the optimal revenue when we do implement p^* .

For the second problem, we set $t_{\max} = 40$. The optimal revenue is equal to 99020.27, and the optimal price vector is implemented from step 33 onwards. The local search algorithm yields a revenue of 98898.67, which is 99.88% of the optimal revenue. The straight line algorithm outputs a revenue of 92192.83, which is 93.11% of optimal. Both these algorithms instantly give the optimal solution. Solving the integer program takes 2 seconds.

Sample 2 Price vector $p^* = (0.109423, 0.173030, 0.216197, 0.666756)$ is optimal for the second sample. The minimum number of time steps needed to implement p^* starting from p^0 is 29. The total revenue generated by the straight line algorithm is 35339.58. The optimal revenue for implementing p^* in 29 steps is 37449.57, which means that the revenue of the straight line algorithm is 94.37% of the optimal revenue. The local search algorithm needs at least 32 steps to implement p^* . Running this algorithm for 29 steps generates a total revenue of 37244.64, which is 99.45% of the optimal revenue when we do implement p^* .

For the second problem, we set $t_{\max} = 40$. The optimal revenue is equal to 56229.31, and the optimal price vector is implemented from step 29 onwards. The local search algorithm yields a revenue of 56004.24, which is 99.60% of the optimal revenue. The straight line algorithm outputs a revenue of 54121.09, which is 96.25% of optimal. Both these algorithms instantly give the optimal solution. Solving the integer program takes 338 seconds.

Sample 3 In the last sample, the optimal price vector is $p^* = (0.112089, 0.188362, 0.102176, 1.467432)$. The minimum number of time steps needed

to implement p^* starting from p^0 is 33. The total revenue generated by the straight line algorithm is 24293.19. The optimal revenue for implementing p^* in 33 steps is 27825.56, which means that the revenue of the straight line algorithm is 87.31% of the optimal revenue. In this sample, the local search algorithm also needs 33 steps to implement p^* , and generates a revenue of 27736.29, which is 99.68% of the optimal revenue.

For the second problem, we set $t_{\max} = 40$. The optimal revenue is equal to 36833.93, and the optimal price vector is implemented from step 33 onwards. The local search algorithm yields a revenue of 36744.66, which is 99.76% of the optimal revenue. The straight line algorithm outputs a revenue of 33301.56, which is 90.41% of optimal. Both these algorithms instantly give the optimal solution. Solving the integer program takes 3 seconds.

	Sample 1		Sample 2		Sample 3	
$\min t_{\max}$	31		29		33	
MIP	70936.90		37449.57		27825.56	
Straight	64230.64	90.55%	35339.58	94.37%	24293.19	87.31%
LS till t_{\max}	70968.47	100.04%	37244.64	99.45%	27736.29	99.68%
t_{\max}	40		40		40	
MIP	99020.27		56229.31		36833.93	
Straight	92192.83	93.11%	54121.09	96.25%	33301.56	90.41%
LS	98898.67	99.88%	56004.24	99.60%	36744.66	99.76%

Table 6.1: Summary of computational results where $p^0 = (0.01, 0.06, 0.01, 0.75)$ and $\delta = 0.05$.

6.5 Conclusion

Regarding Problem 6.1, we can easily calculate the minimum t_{\max} , and find price vectors p^t for all $t \in T^+$ in polynomial time using the straight line algorithm. The local search algorithm might need more steps to implement p^* , but using this algorithm might be more profitable to the company than using the integer program, as there we have the requirement that $p^{t_{\max}}$ should be equal to p^* .

For Problem 6.2 restricted to integer prices, we present a dynamic programming algorithm that runs in pseudo-polynomial time if the number

of item types m is constant. We extend this result and derive a $(1 + \varepsilon)$ -approximation algorithm for the same problem with continuous prices, for any desired precision $\varepsilon > 0$. The local search algorithm seems very useful in practice, and yields close-to-optimal revenues according to our computational results.

Chapter 7

Algorithms for Optimal Price Regulations

Since summer 2007, mobile phone users in the European Union (EU) are protected by a ceiling on the roaming tariff when calling or receiving a call abroad. We analyze the effects of this price regulative policy, and compare it to alternative implementations of price regulations. The problem is a three-level program: The EU determines the price regulative policy, the telephone operator sets profit maximizing prices, and customers choose to accept or decline the operator's offer. The first part of this chapter contains a polynomial time algorithm to solve such a three-level program. The crucial idea is to partition the polyhedron of feasible price regulative parameters into a polynomial number of smaller polyhedra such that a certain primitive decision problem can be written as a linear program on each of those. Then the problem can be solved by a combination of enumeration and linear programming. In Section 7.3, we analyze an instance of this problem more extensively, namely the price regulation problem that the EU encounters. Using customer data from a large telephone operator that resemble reality, we compare different price regulative policies with respect to their social welfare. On the basis of a specific social welfare function, we observe that other price regulative policies or different ceilings can improve the total social welfare on this particular data set.¹

¹This chapter is based on Grigoriev, van Loon, and Uetz (2008a).

7.1 Introduction

It is of major importance to the European Union (EU) that European companies, governments and citizens play an important role at the realization of a world-economy based on knowledge. The EU tries to stimulate the development and use of new information and communication technology, and to enlarge the level of competition of the EU compared to other markets, e.g. the United States and Japan. An important element of the European policy is to assure that ICT-services are available and affordable for everyone. This contains for example telephony, fax, internet and free emergency numbers. However, especially the prices for making and receiving calls abroad, referred to as *roaming*, have been extremely high in the EU. A warning did not lead to a decrease in prices, and therefore the European Commission uses *price regulation* to force lower prices and more transparency in the market (European Commission, Information Society, Media Directorate General 2007). Currently, the EU considers the same instrument for *data roaming*, since the situation mirrors the one for voice roaming back in 2007. We analyze the effects of the current price regulation on the social welfare, both algorithmically and practically, and compare it to alternative regulations.

We regard a general model in which a government tries to maximize *social welfare* through price regulation. This regulation should bound the company (not necessarily a telephone operator) in setting the prices so as to protect the customers, without harming the market. The company determines the price of *item types*. An item type represents a physical product or a service, for example, a minute calling, internet connection, shipping, etc. We use a model in which there is only one company that determines the pricing strategy, under given and known market competition. Obviously, there exist markets in which multiple companies operate and need to share a set of customers. However, for this research we make the simplifying assumption that, under given market regulations, companies reach an optimal market price, so we identify them with just one single company.

Similarly as in Chapter 5, we introduce the following notation. Let $K = \{1, \dots, m\}$ be the set of distinct item types a given company wants to price. Let $p_k \geq 0$ be the price of items of type $k \in K$. Let $J = \{1, \dots, n\}$ be a set of single-minded customers. Every customer $j \in J$ has positive demand d_{jk} for item type $k \in K$, which is the number of times customer j wants to purchase item type k . For example, if k represents calling abroad for one minute then d_{jk} is the number of minutes j wants to call abroad. Or, if k is the start-up

of a process then d_{jk} is 1 if j prefers to start the process and 0 otherwise. Every customer j requests a *contract* from the company, which is specified by the total demand vector (d_{j1}, \dots, d_{jm}) . The valuation b_j is the maximum amount customer $j \in J$ is willing to pay for her contract. Once the prices of the item types are determined, the price of the contract, $p(j)$, is defined by the following *affine function*

$$p(j) = d_{j0} + d_{j1}p_1 + \dots + d_{jm}p_m, \quad j \in J. \quad (7.1)$$

Note that the price of a contract is *personal*, due to a potential ‘entrance fee’ d_{j0} , and because it depends on the demand (d_{j1}, \dots, d_{jm}) of a single customer. We assume that, through market research, we know the customers’ demands d_{jk} for all item types $k \in K$ and the valuation b_j , for all customers $j \in J$. Therefore, we are faced with a purely algorithmic problem in contrast to mechanism design problems where the valuations are private information to the customers.

The pricing regime as defined in (7.1) is referred to as *affine pricing* in the economic literature, which is extensively discussed in Chapter 5. A solution to the problem is a price $p(j)$ for every customer $j \in J$, which is determined through a vector of prices $p = (p_1, \dots, p_m)$ as given in (7.1). Every customer decides whether to accept this contract or not. Hereto, she sets binary variable w_j to 1 if she accepts, and 0 otherwise, in order to maximize her personal objective, denoted by $f_j(p) \cdot w_j$. In this chapter, we assume a linear function $f_j(p)$, for example, think of $f_j(p) = b_j - p(j)$. Let $w = (w_1, \dots, w_n)$ denote the strategies of all customers. Customers that accept the contract are referred to as *winners*, and the set of winners is defined by $W = \{j \in J : w_j = 1\}$. We assume that all item types are available in *unlimited* supply, which is true for digital items for example. Thus any solution will be trivially envy-free.

7.1.1 Three-level program

The government protects the customers by means of regulative constraints. Let $R = \{1, \dots, \hat{r}\}$ denote the set of constraints imposed by the government. Throughout this chapter, we assume that the number of regulations \hat{r} is constant. Every constraint $r \in R$ is defined by $g_r(p, \alpha_r) \leq 0$, where $\alpha = (\alpha_1, \dots, \alpha_{\hat{r}}) \in \mathbb{R}_+^{\hat{r}}$ is a vector of price regulative *parameters* determined by the government. For example, a ceiling on price p_k is implemented by letting $g_k(p, \alpha_k) = p_k - \alpha_k$. We restrict $g_r(p, \alpha_r)$ to be a *linear* function in p and

α . We introduce a bilevel program in which the company maximizes his objective $f^C(p, w)$, for example revenue minus production costs, such that price vector $p = (p_1, \dots, p_m)$ satisfies the price regulative constraints. Every customer maximizes her objective $f_j(p) \cdot w_j$. For further reading on bilevel programs, we refer to Vicente and Calamai (1994) and Marcotte and Savard (2005).

$$\begin{aligned}
 \text{2LEVP : } \max_p \quad & f^C(p, w) \\
 \max_{w_j} \quad & f_j(p) \cdot w_j \quad \forall j \in J \\
 \text{s.t.} \quad & g_r(p, \alpha_r) \leq 0 \quad \forall r \in R \\
 & p_k \geq 0 \quad \forall k \in K \\
 & w_j \in \{0, 1\} \quad \forall j \in J
 \end{aligned}$$

In the above mathematical program, the vector of price regulative parameters α and constraints $g_r(p, \alpha_r)$, $r \in R$, are assumed to be given. We propose a *three-level* program², where, on top of the two levels given by company and customers, the government maximizes social welfare. In this three-level program, the government acts first, the company decides given the government's action, and the customers make their decision based on the company's prices. The government influences social welfare by modifying the price regulative parameters α_r , $r \in R$. Also, instead of simply forbidding the violation of price regulative constraints, we introduce taxes $\tau = (\tau_1, \dots, \tau_{\hat{r}}) \in [0, 1]^{\hat{r}}$. That is, if the company's prices violate regulation $g_r(p, \alpha_r) \leq 0$ then a penalty must be paid over the additional revenue received by this violation. The actual penalty for violating regulation r is denoted by

$$f_r^{\text{TAX}}(p, w, \alpha, \tau) = g_r(p, \alpha_r)^+ \cdot \tau_r \cdot \bar{g}_r(w),$$

where $g_r(p, \alpha_r)^+ = \max\{0, g_r(p, \alpha_r)\}$ is the amount of violation, τ_r is the tax, and function $\bar{g}_r(w)$ needs to be specified for each type of price regulative constraint r . For example, if the regulation is a ceiling on p_1 , that is, $g_r(p, \alpha_r) = p_1 - \alpha_r$, it sounds reasonable to ask tax τ_r for each euro earned by violating the ceiling. Then $\bar{g}_r(w)$ would have to be defined as the total demand for item type 1 of all winners, $\sum_{j \in J} d_{j1} w_j$.

Let us denote the total tax payment of the company by

$$f^{\text{TAX}}(p, w, \alpha, \tau) = \sum_{r \in R} f_r^{\text{TAX}}(p, w, \alpha, \tau) = \sum_{r \in R} g_r(p, \alpha_r)^+ \cdot \tau_r \cdot \bar{g}_r(w). \quad (7.2)$$

²This three-level program is loosely based on Courcoubetis and Weber (2003, Section 5.4.1), but there the model does not include price regulation.

The mathematical program (3LEVP) below shows the general structure of the three-level program that we consider.

$$\begin{array}{llll}
\text{3LEVP : } \max_{\alpha, \tau} & f^G(p, w) & & \\
\max_p & f^C(p, w) - f^{\text{TAX}}(p, w, \alpha, \tau) & & \\
\max_{w_j} & f_j(p) \cdot w_j & \forall j \in J & \\
& w_j \in \{0, 1\} & \forall j \in J & \\
& p_k \geq 0 & \forall k \in K & \\
& 0 \leq \tau_r \leq 1 & \forall r \in R & \\
& \alpha_r \geq 0 & \forall r \in R. &
\end{array}$$

We impose the following conditions on the functions.

Condition 7.1. *The government's objective function $f^G(p, w)$ is a polynomially computable function.*

Condition 7.2. *For every customer $j \in J$, the function $f_j(p)$ is linear in p .*

Condition 7.3. *Given w , α and τ , function $f^C(p, w) - f^{\text{TAX}}(p, w, \alpha, \tau)$ is positive, piecewise linear and monotone in p .*

Condition 7.4. *Price regulative constraint $g_r(p, \alpha_r)$ is linear in p and α_r for all $r \in R$.*

The objective of the government could be to maximize the social welfare. This objective is modeled by function $f^G(p, w)$ in the above three-level program. If the above conditions are satisfied then we show that we can solve this program in polynomial time, given that the number of item types m and the number of regulative constraints \hat{r} are constant.

Note that the lower two levels of 3LEVP can be seen as the Lagrangian of the bilevel program 2LEVP. Moreover, a strict price regulation as in the bilevel model, that is, forbidding to violate the regulative constraints, can still be implemented by letting τ_r be arbitrarily large for all $r \in R$. In this case, τ is not strictly a tax, but just an indication that we forbid to violate the price regulation constraints.

It is known, as shown by Demaine et al. (2006), that problem 2LEVP with a non-constant number of item types m is hard to approximate within a semi-logarithmic factor in the number of customers n . This means that solving the three-level program 3LEVP has the same complexity already if

the government's objective is equal to the company's objective. Similarly as in Chapter 5, we make the assumption that the number of distinct item types m is constant. These are reasonable assumptions particularly for the applications that we aim at, since there the number of item types is very small, for example, price per minute for a call received, a call placed, and price per text message. For a small number of item types, the number of regulations is also naturally assumed to be small.

For a constant number of item types m and regulative constraints \hat{r} , we present a polynomial time algorithm to solve three-level program 3LEVP, under the four conditions described above.

In Section 7.3, we explicitly define all functions in three-level program 3LEVP to optimize the social welfare for the specific problem faced by the European Union in regulating the roaming charges. We present polynomial time algorithms to find the optimal social welfare for the current EU policy, optimization of the tax level only, and optimization of the regulative parameters only. After the theoretical results and description of the algorithms, we perform an extensive computational study in Section 7.4 to verify practical feasibility of the approach and to evaluate the results of different scenarios for implementing price regulations. Here, we use the actual price regulations set by the European Commission and investigate the EU policy in terms of social welfare.

7.2 Parameter and tax level optimization

Consider three-level program 3LEVP for which the functions satisfy the conditions stated before. In this section, we propose an algorithm to solve this program.

Definition 7.2.1 (Vertex). *A vertex $v \in V$ is defined by a price vector $p^{(v)} \in \mathbb{R}_+^m$ that is characterized by m equalities out of the $n + m$ equalities $f_j(p^{(v)}) = 0$, $j \in J$, and $p_k^{(v)} = 0$, $k \in K$, such that the coefficient matrix of these m equalities has rank m .*

Lemma 7.2.2. *For given vectors $w \in \{0, 1\}^n$, $\alpha \in \mathbb{R}_+^{\hat{r}}$ and $\tau \in [0, 1]^{\hat{r}}$, the optimal price vector $p = (p_1, \dots, p_m)$ for the company can only be at a vertex as defined in Definition 7.2.1.*

Proof. The lemma follows from the fact that, for given vectors α and τ , the company's objective function is *locally linear* in the prices. Here, locally

linear means linear on any minimal polyhedral subdivision as defined by the $n + m + \hat{r}$ equations $f_j(p) = 0$, $j \in J$, $p_k = 0$, $k \in K$, and $g_r(p, \alpha_r) = 0$, $r \in R$. Thus, any price vector p that is not a vertex either has an improving direction since $\tau_r \leq 1$ for all $r \in R$ and therefore is not optimal, or it lies on a face of the arrangement of the m linear equations that is orthogonal to the linear objective. Moreover, since $\tau_r \leq 1$ for all $r \in R$, moving in the improving direction will never decrease the company's profit, no matter what α is. \square

We propose an algorithm that solves problem 3LEVP by optimizing price regulative parameters α and the taxes τ simultaneously. Thereto, we partition the polyhedron of the price regulative parameters α into a polynomial number of smaller polyhedra, and solve a linear programming problem in each of those. These linear programs are defined in such a way that we can verify if a given price vector constitutes the company's optimum prices. This decision problem is in general non-linear. The trick here is to define the partition such that this decision problem becomes linear inside each of the small polyhedra. The optimum solution of a three-level mathematical program is eventually obtained by enumeration over all polyhedra and vertices, and evaluating the social welfare in each of them. We first describe the algorithm formally, and then illustrate our approach in Example 7.2.4.

Consider the following decision problem.

Problem 7.1. *Are there price regulative vectors α and τ such that vertex $v \in V$ with price vector $p^{(v)}$ maximizes the objective function $f^C - f^{\text{TAX}}$ for the company?*

The main idea for the solution to Problem 7.1 is as follows. Consider some arbitrary vertex $v \in V$, we would like to write constraints expressing the fact that vertex v maximizes the objective, namely for all vertices $u \in V$,

$$f^C(p^{(v)}, w^{(v)}) - f^{\text{TAX}}(p^{(v)}, w^{(v)}, \alpha, \tau) \geq f^C(p^{(u)}, w^{(u)}) - f^{\text{TAX}}(p^{(u)}, w^{(u)}, \alpha, \tau).$$

By definition, $f^{\text{TAX}}(p^{(v)}, w^{(v)}, \alpha, \tau)$ is nonlinear in the price regulative parameters. To linearize f^{TAX} we introduce a subdivision of $\mathbb{R}_+^{\hat{r}}$ into polyhedra A_ℓ , $\ell \in L$. Thus, L is the set of polyhedra. For a given vertex $v \in V$ and by linearity of $g_r(p^{(v)}, \alpha_r)$, there is a unique value for α_r , say α_r^v , where the sign of $g_r(p^{(v)}, \alpha_r)$ changes from ≤ 0 to > 0 . Doing this for all vertices $v \in V$ and all regulative constraints $r \in R$, we define a rectangular subdivision in $\mathbb{R}_+^{\hat{r}}$ for possible α 's by $\alpha_r = \alpha_r^v$. On each such defined α -rectangle A_ℓ , we may now compare the company's objective in vertex $v \in V$ to the objective in all

other vertices $u \in V \setminus v$. Therefore, we split the tax function f_r^{TAX} , $r \in R$, into two parts for every vertex $v \in V$ and polyhedron A_ℓ such that $x_r^{(v,\ell)}$ incorporates all terms in the regulative constraint not multiplied with α_r and $y_r^{(v,\ell)}$ incorporates all terms multiplied with α_r . More precisely, let $\check{\alpha}_r^{(\ell)}$ and $\hat{\alpha}_r^{(\ell)}$ be the lower and upper bound for α_r in polyhedron A_ℓ . For every vertex $v \in V$, polyhedron A_ℓ , $\ell \in L$, and regulation $r \in R$, we do the following. If $g_r(p^{(v)}, \hat{\alpha}_r^{(\ell)}) \leq 0$ then let coefficients $x_r^{(v,\ell)} = y_r^{(v,\ell)} = 0$. Otherwise, that is, if $g_r(p^{(v)}, \hat{\alpha}_r^{(\ell)}) > 0$ then

$$x_r^{(v,\ell)} = g_r(p^{(v)}, 0) \cdot \bar{g}(w^{(v)}) \text{ and } y_r^{(v,\ell)} = \frac{\partial g_r}{\partial \alpha_r}(p^{(v)}, \alpha_r) \cdot \bar{g}(w^{(v)}).$$

Note that these definition of $x_r^{(v,\ell)}$ and $y_r^{(v,\ell)}$ are correct because $g_r(p^{(v)}, \alpha)$ is linear in α . Consequently,

$$f_r^{\text{TAX}}(p^{(v)}, w^{(v)}, \alpha, \tau) = g_r(p^{(v)}, \alpha_r)^+ \cdot \tau_r \cdot \bar{g}_r(w^{(v)}) = x_r^{(v,\ell)} \tau_r + y_r^{(v,\ell)} \alpha_r \tau_r, \quad \forall r \in R.$$

We can derive a solution to Problem 7.1 on A_ℓ by solving the following mathematical program for every vertex $v \in V$.

$$\begin{aligned} & f^C(p^{(v)}, w^{(v)}) - \left(\sum_{r \in R} x_r^{(v,\ell)} \tau_r + y_r^{(v,\ell)} \alpha_r \tau_r \right) \\ & \geq f^C(p^{(u)}, w^{(u)}) - \left(\sum_{r \in R} x_r^{(u,\ell)} \tau_r + y_r^{(u,\ell)} \alpha_r \tau_r \right) \quad \forall u \in V \\ & \check{\alpha}_r^{(\ell)} \leq \alpha_r \leq \hat{\alpha}_r^{(\ell)} \quad \forall r \in R \\ & 0 \leq \tau_r \leq 1 \quad \forall r \in R. \end{aligned}$$

This quadratic program can be linearized by simple variable substitution, where $\phi_r = \alpha_r \tau_r$ for all $r \in R$. Therefore, for every vertex $v \in V$ and polyhedron A_ℓ , Problem 7.1 becomes a linear program (LP1) with variables τ_r and ϕ_r for all $r \in R$.

$$\begin{aligned} \text{LP1: } & \sum_{r \in R} \left(x_r^{(v,\ell)} - x_r^{(u,\ell)} \right) \tau_r + \left(y_r^{(v,\ell)} - y_r^{(u,\ell)} \right) \phi_r \\ & \leq f^C(p^{(v)}, w^{(v)}) - f^C(p^{(u)}, w^{(u)}) \quad \forall u \in V \\ & \check{\alpha}_r^{(\ell)} \tau_r \leq \phi_r \leq \hat{\alpha}_r^{(\ell)} \tau_r \quad \forall r \in R \\ & 0 \leq \tau_r \leq 1 \quad \forall r \in R. \end{aligned}$$

On any polyhedron A_ℓ this linear program is either infeasible, suggesting that there are no price regulative parameters in A_ℓ that makes v the solution that maximizes the company's objective, or we obtain corresponding

price regulative parameters in A_ℓ . Eventually, a straightforward algorithm enumerating all vertices $v \in V$, checking feasible solutions for α 's in A_ℓ , $\ell \in L$, and picking the one that achieves the maximal social welfare, provides an optimal solution to the three-level program. Since the number of item types m is constant, we have a polynomial number of vertices. As the number of regulative constraints \hat{r} is constant, we have only a polynomial number of polyhedra in $\mathbb{R}_+^{\hat{r}}$. For every polyhedron and every vertex, we solve linear program LP1, deriving the following theorem.

Theorem 7.2.3. *Three-level program 3LEVP, satisfying the imposed conditions, admits a polynomial time algorithm if the number of item types m and the number of regulative constraints \hat{r} are constant.*

Example 7.2.4. *Consider the problem in which $m = 2$ with $f_j(p) = b_j - p(j)$ for all $j \in J$. Figure 7.1 displays a possible picture for this setting, where solid lines (1), (2) and (3) represent the equalities $f_j(p) = 0$ of three customers.*

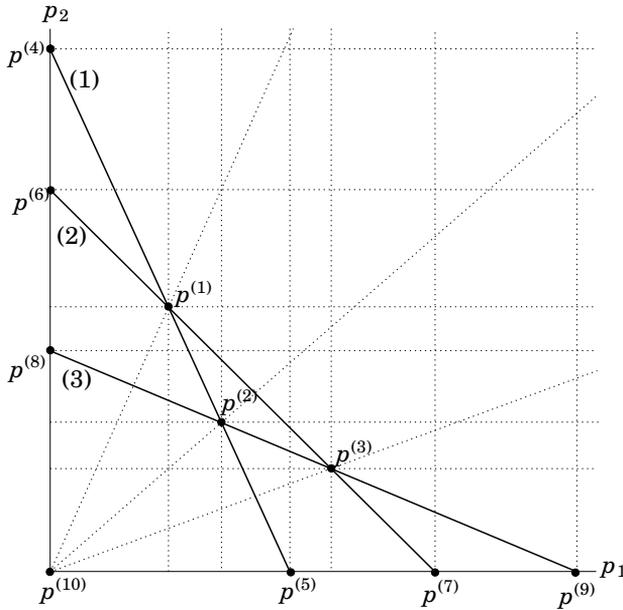


Figure 7.1: Linear arrangement of the valuation and nonnegativity inequalities.

The vertices $p^{(1)}, \dots, p^{(10)}$ generated by m out of these equalities and the equalities $p_k = 0$, $k \in K$, are displayed by the bullets. We define three price regulative constraints by $g_1(p, \alpha_1) = p_1 - \alpha_1$, $g_2(p, \alpha_2) = p_2 - \alpha_2$ and

$g_3(p, \alpha_3) = p_1 - \alpha_3 p_2$, and let $\bar{g}_r(w) = 1$ for all $r = 1, 2, 3$. There is no economic interpretation for $\bar{g}_r(w) = 1$, so this is purely for simplicity of the example. These constraints are presented by the dotted lines in Figure 7.1, created by solving $g_r(p, \alpha_r) = g_r(p^{(v)}, \alpha_r) = 0$ for all $v \in V$ and $r \in R$. That is, the vertical, horizontal and diagonal lines represent regulative constraint g_1 , g_2 and g_3 , respectively, through the different vertices. There are 7 vertical regions, that is, from $p_1 = 0$ to $p_1 = p_1^{(1)}$, from $p_1 = p_1^{(1)}$ to $p_1 = p_1^{(2)}$, etcetera, all the way up to $p_1 \geq p_1^{(9)}$. Similarly, there are 7 horizontal regions, and 4 diagonal regions. Consequently, we need $7 \cdot 7 \cdot 4 = 196$ polyhedra in the subdivision of the regulative parameter space \mathbb{R}_+^3 .

Figure 7.2 represents the subdivision of the regulative parameter space \mathbb{R}_+^3 into polyhedra A_ℓ , $\ell = 1, \dots, 196$. On the axes, there are the values for the α 's on the vertices.

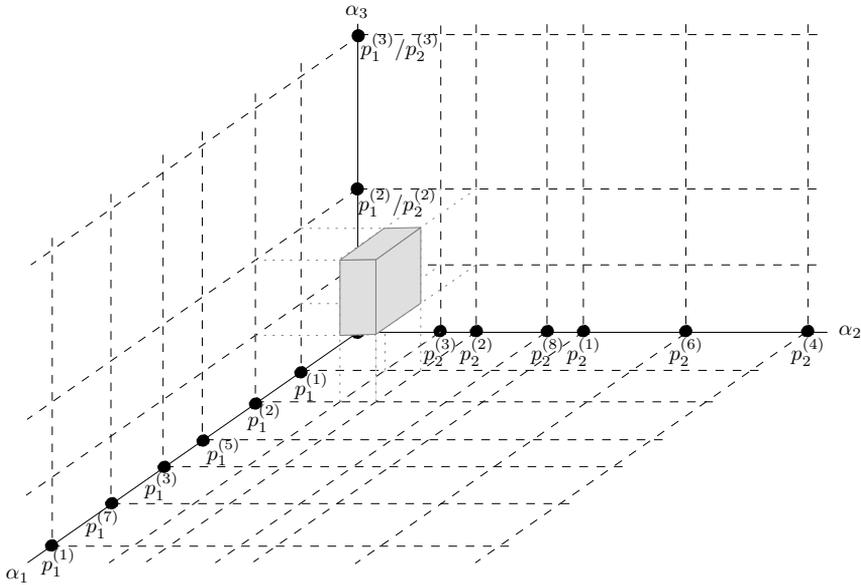


Figure 7.2: Subdivision in \mathbb{R}_+^3 .

As an example, consider the gray colored polyhedron A_ℓ . This polyhedron is defined by

$$\begin{aligned}
 p_1^{(1)} = \check{\alpha}_1^{(\ell)} &\leq \alpha_1 \leq \hat{\alpha}_1^{(\ell)} = p_1^{(2)} \\
 p_2^{(3)} = \check{\alpha}_2^{(\ell)} &\leq \alpha_2 \leq \hat{\alpha}_2^{(\ell)} = p_2^{(2)} \\
 p_1^{(1)}/p_2^{(1)} = \check{\alpha}_3^{(\ell)} &\leq \alpha_3 \leq \hat{\alpha}_3^{(\ell)} = p_1^{(2)}/p_2^{(2)}.
 \end{aligned}$$

Now, let us consider vertex $p^{(3)}$. As $g_1(p^{(3)}, \hat{\alpha}_1^{(\ell)}) > 0$, we have $x_1^{(3,\ell)} = p_1^{(3)}$ and $y_1^{(3,\ell)} = -1$. For the second regulation we have $g_2(p^{(3)}, \hat{\alpha}_2^{(\ell)}) < 0$, thus $x_2^{(3,\ell)} = y_2^{(3,\ell)} = 0$. And finally, since $g_3(p^{(3)}, \hat{\alpha}_3^{(\ell)}) > 0$ we have $x_3^{(3,\ell)} = p_1^{(3)}$ and $y_1^{(3,\ell)} = -p_2^{(3)}$. We perform these steps for every polyhedron A_ℓ , $\ell \in L$, and every vertex $v \in V$. After straightforward computation of $f^C(p^{(u)}, w^{(u)})$, $u \in V$, we solve linear program LP1 and select the vertex that yields the highest social welfare among all vertices $v \in V$ for which LP1 is feasible.

In this section, we use price regulative constraints in which there is one particular α_r and τ_r for every constraint $r \in R$. However, the above described algorithm can easily be adapted for other cases, that is, there might be multiple α parameters or tax levels τ in one constraint, or multiple constraints can be subjected to the same tax level τ or contain the same α .

7.3 Optimization of European regulation

In this section, we explicitly define functions f^G , f^C , f^{TAX} and f_j , $j \in J$, to address the problem faced by the European Union regarding the regulation on roaming. First of all, the customers accept a contract if its price does not exceed the valuation, that is, the objective of customer $j \in J$ is defined as $(b_j - p(j))w_j$. If $p(j) \leq b_j$ this function is maximized by setting $w_j = 1$ and thus accepting the contract, and 0 otherwise. Furthermore, the EU sets ceilings on the prices. That is, $R = K$, and $g_k(p, \alpha_k) = p_k - \alpha_k$ with $\alpha_k \geq 0$ for all item types $k \in K$. The objective of the company is to maximize the profit, defined as revenue minus costs. The revenue is the total payment by all winning customers. As for the costs, let c_k be the cost of providing one unit of k to a customer. The company's cost to serve customer j is denoted by $c(j) = d_{j1}c_1 + \dots + d_{jm}c_m$.

In the simplest setting, the government regulates the prices by forbidding to violate the constraint $p_k \leq \alpha_k$ for every item type $k \in K$. We model this by the following bilevel program.

$$\begin{array}{ll}
 \max_p & \sum_{j \in J} (p(j) - c(j))w_j \\
 \max_{w_j} & (b_j - p(j))w_j \\
 \text{s.t.} & 0 \leq p_k \leq \alpha_k \quad \forall k \in K \\
 & w_j \in \{0, 1\} \quad \forall j \in J.
 \end{array}$$

To find the prices that will lead to the optimal profit for the company, we use the affine pricing algorithm introduced in Chapter 5, in which we incorporate the price regulative constraints, and the costs of the company.

Theorem 7.3.1 (Theorem 5.2.5). *For given price regulative constraints $p_k \leq \alpha_k$ which must not be violated, profit maximizing prices can be computed in polynomial time, given that the number of distinct item types m is constant.*

As already discussed in the introduction, we study whether there are other price regulative strategies that might lead to an increase in *social welfare*. Thus, let us first proceed with a definition of the social welfare function we believe is appropriate for modeling the roaming regulation problem. According to utilitarians such as Jeremy Bentham³ and John Stuart Mill⁴, society should aim to maximize the total *utility* of individuals, aiming for “the greatest happiness for the greatest number”. Thus, the government strives to set the price regulative parameters so as to maximize the social welfare, defined as the sum of utilities. The utility of the company is the total revenue minus costs. We assume that the company is risk-neutral, and thus the marginal utility is equal for every extra euro earned. On the contrary, consumers have a concave utility function in general, which means that they are assumed to be risk-averse. A concave utility function induces that a gain in wealth conveys a smaller increase to utility than the reduction in utility imparted by a loss in wealth of equal magnitude, that is, *diminishing marginal utility*. Another property of a concave utility function is that a customer with a low valuation values one unit of money more than a customer with a high valuation. In other words, the marginal utility of a euro to a ‘poor’ customer is likely to exceed the marginal utility of a euro to a ‘rich’ customer (Shavel 2004, Chapter VII). Bernoulli (1738)⁵ first proposed a utility function that is equal to the natural logarithm of wealth. A logarithmic function is monotonically increasing and the marginal utility function is monotonically decreasing, which are the two basic mathematical properties that consumer utility functions have to satisfy, as described by Baz and Chacko (2004). In the words of Savage (1972), “no other function has been

³Jeremy Bentham (1748-1832) was an English jurist, philosopher, and legal and social reformer.

⁴John Stuart Mill (1806-1876) was a British philosopher, political economist, civil servant and Member of Parliament.

⁵Daniel Bernoulli (1700-1782) was a Swiss mathematician. He is particularly remembered for his applications of mathematics to mechanics, especially fluid mechanics, and for his pioneering work in probability and statistics.

suggested as a better prototype for Everyman’s utility function”. Based on this discussion, we model the utility of customer $j \in J$ as $\ln((b_j - p(j))w_j + 1)$, where the addition of 1 is solely to have a positive function. Conclusively, the social welfare, and thus the government’s objective function, is defined as

$$f^G = \sum_{j \in J} \ln((b_j - p(j))w_j + 1) + (p(j) - c(j))w_j.$$

In Section 7.3.1, we furthermore include a tax payment for violating the price regulative constraints, similar as in Section 7.2. However, this does not change the social welfare function, as the tax is paid by the company to the government. Thus, the company’s utility is decreased by the same amount as the government’s utility is increased, also known as *transferable utility*. This assumption is justified when the company and the government have a common currency that is valued equally by both.

7.3.1 Price regulation by tax

In this section, the price regulative constraints $p_k \leq \alpha_k$ are not enforced by law, but their violation is penalized via tax τ_1 . We define $\bar{g}_k(w) = \sum_{j \in J} d_{jk}w_j$ for all item types $k \in K$. The tax function is defined as

$$f_k^{\text{TAX}}(p, w, \alpha, \tau_1) = g_k(p, \alpha_k)^+ \cdot \tau_1 \cdot \bar{g}_k(w) = \sum_{j \in J} d_{jk}(p_k - \alpha_k)^+ \tau_1 w_j, \quad \forall k \in K.$$

Thus, given vector α , the government determines tax level τ_1 to maximize social welfare.

$$\begin{aligned} \max_{\tau_1} \quad & f^G = \sum_{j \in J} \ln((b_j - p(j))w_j + 1) + (p(j) - c(j))w_j \\ \max_p \quad & f^C - f^{\text{TAX}} = \sum_{j \in J} \left(p(j) - c(j) - \sum_{k \in K} d_{jk}(p_k - \alpha_k)^+ \tau_1 \right) w_j \\ \max_{w_j} \quad & f_j w_j = (b_j - p(j))w_j \\ \text{s.t.} \quad & p_k \geq 0 \quad \forall k \in K \\ & w_j \in \{0, 1\} \quad \forall j \in J \\ & 0 \leq \tau_1 \leq 1. \end{aligned}$$

Notice that all functions in this three-level program satisfy the conditions stated in the introduction of this chapter.

For any given $\alpha_k, k \in K$, consider the arrangement of linear inequalities defined in \mathbb{R}_+^m by the *valuation inequalities* $p(j) \leq b_j$ for every customer $j \in$

J , that is, $f_j(p) \geq 0$, nonnegativity inequalities $p_k \geq 0$ and price regulative constraints $p_k \leq \alpha_k$ for every item type $k \in K$.

Definition 7.3.2 (Vertex). *A vertex $v \in V$ is defined by a price vector $p^{(v)} = (p_1^{(v)}, \dots, p_m^{(v)}) \in \mathbb{R}_+^m$ that satisfies m equalities out of the $n + 2m$ equalities $p^{(v)}(j) = b_j$, $j \in J$, $p_k^{(v)} = \alpha_k$, and $p_k^{(v)} = 0$, $k \in K$, such that the coefficient matrix of these m equalities has rank m .*

To optimize the social welfare function we use nonlinear programming. For further reading on nonlinear programming, we refer to Bazaraa, Sherali, and Shetty (1993). For notational purposes, let $I = \{1, \dots, n + 2m\}$ be the *index set of inequalities*. For all $i \in I$, we do the following:

- If $i = j \in J$ then $d_{ik} = d_{jk}$ for all $k \in K \cup 0$ such that $p(i) = p(j)$ and $b_i = b_j$.
- If $i - n = k \in K$ then $d_{ik} = 1$ and $d_{ik'} = 0$ for all $k' \in (K \cup 0) \setminus k$ such that $p(i) = p_k$, and $b_i = 0$.
- If $i - n - m = k \in K$ then $d_{ik} = 1$ and $d_{ik'} = 0$ for all $k' \in (K \cup 0) \setminus k$ such that $p(i) = p_k$, and $b_i = \alpha_k$.

Furthermore, we define

$$S(p) = \sum_{j \in W} \ln(b_j - p(j) + 1) + p(j) - c(j),$$

for some given price vector $p \in \mathbb{R}_+^m$, where $W = \{j \in J : p(j) \leq b_j\}$.

Consider the arrangement of linear inequalities $p(i) \leq b_i$. This arrangement consists of multiple polyhedra. A polyhedron \mathcal{P} is defined by a subset of inequalities $U_i(p) \geq 0$, where $U_i(p) = p(i) - b_i$ for some $i \in I$ and $U_i(p) = b_i - p(i)$ for others.

Example 7.3.3. *Consider an instance with $n = 3$ customers and $m = 2$ item types, thus $I = \{1, \dots, 7\}$. The arrangement of linear inequalities $p(i) \leq b_i$ displayed in Figure 7.3. The gray polyhedron is defined by $U_2(p) = b_2 - p(2) \geq 0$, $U_3(p) = p(3) - b_3 \geq 0$ and $U_6(p) = p(6) - b_6 = p_1 - \alpha_1 \geq 0$.*

The Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for a solution to be optimal. The necessary conditions for inequality constrained problems were first published by Karush (1939), although they became renowned after a seminal conference paper by Kuhn and Tucker

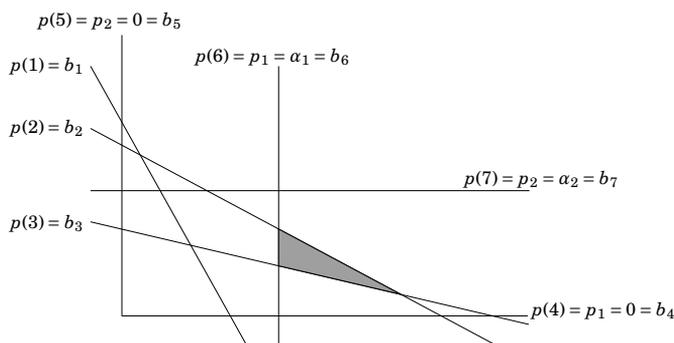


Figure 7.3: A polyhedron in the arrangement of linear inequalities.

(1951). Let $\bar{p} = (\bar{p}_1, \dots, \bar{p}_m)$ be a feasible solution and let $I^\ominus = \{i \in I : U_i(\bar{p}) = 0\}$. If \bar{p} is a local optimum then for $i \in I^\ominus$ unique scalars μ_i exist such that

$$\begin{aligned} \nabla S(\bar{p}) + \sum_{i \in I^\ominus} \mu_i \nabla U_i(\bar{p}) &= 0 \\ \mu_i U_i(\bar{p}) &= 0 & \forall i \in I^\ominus \\ \mu_i &\geq 0 & \forall i \in I^\ominus. \end{aligned}$$

Using the above discussion, we now explicitly show that the optimal pricing strategy is attained at a vertex.

Theorem 7.3.4. *For any given $\alpha_k \geq 0$, $k \in K$, there exists a revenue maximizing price vector $p \in \mathbb{R}_+^m$ corresponding to a vertex as in Definition 7.3.2.*

Proof. Consider an arbitrary chosen polyhedron \mathcal{P} of the arrangement of linear inequalities. We show that for any such polyhedron, the optimal solution can only be attained at a vertex. First, we assume there exists a feasible price vector $\bar{p} = (\bar{p}_1, \dots, \bar{p}_m)$ in the interior of polyhedron \mathcal{P} . As \bar{p} is in the interior, we know that $I^\ominus = \emptyset$. The set of winners is denoted by $W \subseteq J$. For any price vector in the interior of polyhedron \mathcal{P} , the set of winners W is the same. The KKT conditions therefore hold if

$$\nabla S(\bar{p}) = 0 \Leftrightarrow \sum_{j \in W} \frac{-d_{jk}}{b_j - \bar{p}(j) + 1} + d_{jk} = 0, \forall k \in K,$$

where $\bar{p}(j) = d_{j0} + d_{j1}\bar{p}_1 + \dots + d_{jm}\bar{p}_m$. As \bar{p} is in the interior, we know that $b_j > \bar{p}(j)$ for all $j \in W$. Thus $\nabla S(\bar{p}) > 0$, and \bar{p} is not a local optimum.

Now we check whether a price vector on the boundary of the polyhedron can be a local optimum, where we exclude the vertices. We consider two

cases as the inequality $U_i(p) \geq 0$ describing the polyhedron can be either $U_i(p) = p(i) - b_i$ or $U_i(p) = b_i - p(i)$, for some $i \in I$.

Case 1: For some $i \in I$, we assume that $\bar{p}(i) = b_i$ and $\tilde{p}(i) > b_i$ for any \tilde{p} in the interior of polyhedron \mathcal{P} . That is, inequality $U_i(p) = p(i) - b_i \geq 0$ partially describes the polyhedron. Obviously, $I^- = \{i\}$. Then, vector \bar{p} is a local optimum if $(\bar{p}(i) - b_i)\mu_i = 0$, $\mu_i \geq 0$ and

$$\sum_{j \in W} \frac{-d_{jk}}{b_j - \bar{p}(j) + 1} + d_{jk} + \mu_i d_{ik} = 0. \quad (7.3)$$

If $i = j \in W$ then for this particular customer we have $b_j = p(j)$. Note that we may assume that $|W \setminus I^-| \geq 1$ as if the only winner is $i = j$ then we can choose any point on $p(j) = b_j$ and get revenue b_j , so we will obtain the same revenue at a vertex of the polyhedron that is partly defined by $p(j) = b_j$. For all $j \in W \setminus I^-$, it holds that $b_j > \bar{p}(j)$. Coefficient d_{ik} is equal to $d_{jk} \geq 0$ if $i = j \in J$, or 1 if $i - n = k \in K$ or $i - n - m = k \in K$. Also, we know that μ_i is positive for all $i \in I$ and $k \in K$. Thus, the left hand side of Equality (7.3) is strictly positive, and therefore the equality cannot hold, which means that vector \bar{p} is not a local optimum.

Case 2: For some $i \in I$, we assume that $\bar{p}(i) = b_i$ and $\tilde{p}(i) < b_i$ for any \tilde{p} in the interior of polyhedron \mathcal{P} . That is, inequality $U_i(p) = b_i - p(i) \geq 0$ partially describes the polyhedron. Again, $I^- = \{i\}$. By definition, there exists some item type $k' \in K$ such that $d_{ik'} > 0$ and therefore we have

$$\bar{p}_{k'} = \frac{b_i - d_{i0}}{d_{ik'}} - \sum_{k \in K \setminus k'} \frac{d_{ik}}{d_{ik'}} \bar{p}_k.$$

Now, we rewrite social welfare function $S(\bar{p})$ by replacing $\bar{p}_{k'}$ such that the function only contains prices \bar{p}_k , $k \in K \setminus k'$. Consequently, the partial derivative of $S(\bar{p})$ with respect to $\bar{p}_{k'}$ is equal to 0. Vector \bar{p} is a local optimum if

$$\frac{\partial S(\bar{p})}{\partial \bar{p}_{k'}} + \mu_i \frac{\partial U_i(\bar{p})}{\partial \bar{p}_{k'}} = 0 - \mu_i d_{ik'} = 0, \text{ and} \quad (7.4)$$

$$\frac{\partial S(\bar{p})}{\partial \bar{p}_k} + \mu_i \frac{\partial U_i(\bar{p})}{\partial \bar{p}_k} = 0, \forall k \in K \setminus k'. \quad (7.5)$$

Equation (7.4) can only hold if $\mu_i = 0$ as $d_{ik'} > 0$ by definition. Equation (7.5) can be written as

$$\sum_{j \in W} \left(\frac{d_{jk'} \frac{d_{ik}}{d_{ik'}} - d_{jk}}{b_j - \bar{p}(j) + 1} - d_{jk'} \frac{d_{ik}}{d_{ik'}} + d_{jk} \right) - \mu_i d_{ik} = 0,$$

for all $k \in K \setminus k'$. Similarly as before, we may assume that $|W \setminus I^-| \geq 1$. As $b_j > \bar{p}(j)$ for all $j \in W \setminus I^-$, the left hand side of the this latter equality is strictly negative. Thus vector \bar{p} cannot be a local optimum. Concluding, a local optimum can only exists at a vertex of the linear arrangement. \square

Notice that a vertex $v \in V$ is most preferable to the company if the profit after tax at this vertex, $f^C(p^{(v)}, w^{(v)}) - f^{\text{TAX}}(p^{(v)}, w^{(v)}, \alpha, \tau_1)$, is at least as high as at any other vertex. Let $W^{(v)} = \{j \in J : w_j^{(v)} = 1\}$. More precisely, for every vertex $u \in V \setminus v$, the tax level τ_1 must be such that

$$\begin{aligned} \sum_{j \in W^{(v)}} p^{(v)}(j) - c(j) - \sum_{k \in K} d_{jk} (p_k^{(v)} - \alpha_k)^+ \tau_1 &\geq \\ \sum_{j \in W^{(u)}} p^{(u)}(j) - c(j) - \sum_{k \in K} d_{jk} (p_k^{(u)} - \alpha_k)^+ \tau_1. &\quad (7.6) \end{aligned}$$

Note that all terms except τ_1 in the above inequality are known, as α_k is given and $p_k^{(v)}$ is defined for all $k \in K$ and $v \in V$. Let us denote

$$T^{(v)} = \sum_{j \in W^{(v)}} \sum_{k \in K} d_{jk} (p_k^{(v)} - \alpha_k)^+.$$

We rewrite inequality (7.6) and solve the following feasibility linear program (LP2) for each vertex $v \in V$.

$$\begin{aligned} \text{LP2 : } (T^{(v)} - T^{(u)})\tau_1 &\leq \sum_{j \in W^{(v)}} (p^{(v)}(j) - c(j)) - \sum_{j \in W^{(u)}} (p^{(u)}(j) - c(j)) \quad \forall u \in V \setminus v \\ 0 &\leq \tau_1 \leq 1. \end{aligned}$$

Let $V^* \subseteq V$ be the set of vertices for which the above linear program has a feasible solution. Then, among all vertices in V^* we select the vertex v^* with the highest social welfare

$$\sum_{j \in W^{(v^*)}} \ln(b_j - p^{(v^*)}(j) + 1) + p^{(v^*)}(j) - c(j).$$

The tax level τ_1 is obtained as a solution to the linear program for this particular vertex v^* . So we have proved the following theorem.

Theorem 7.3.5. *For given price regulative constraints $p_k \leq \alpha_k$, the tax level $\tau_1 \in [0, 1]$ that maximizes the total social welfare, and the corresponding profit maximizing prices can be computed in polynomial time, given that the number of distinct item types m is constant.*

7.3.2 Parameter optimization

So far we assumed *given* values of the price regulative parameters $\alpha_1, \dots, \alpha_m$. In this section we optimize these parameters under the regulation that the company sets the price $p_k \leq \alpha_k$ for all item types $k \in K$. Hereto, we use the following model.

$$\begin{array}{ll}
 \max_{\alpha} & \sum_{j \in J} (\ln((b_j - p(j))w_j + 1) + (p(j) - c(j))w_j) \\
 \max_p & \sum_{j \in J} (p(j) - c(j))w_j \\
 \max_{w_j} & (b_j - p(j))w_j \quad \forall j \in J \\
 \text{s.t.} & 0 \leq p_k \leq \alpha_k \quad \forall k \in K \\
 & w_j \in \{0, 1\} \quad \forall j \in J \\
 & \alpha_k \geq 0 \quad \forall k \in K.
 \end{array}$$

Since parameters $\alpha_k, k \in K$, are not given, let V denote the set of vertices as defined in Definition 7.2.1. For every vertex $v \in V$, let $\alpha_k = p_k^{(v)}$ for all $k \in K$. Let $U = \{u \in V : p_k \leq \alpha_k, \forall k \in K\}$. Then, vertex $u \in U$ is most preferable to the company if the profit

$$\sum_{j \in W^{(u)}} p^{(u)}(j) - c(j) \geq \sum_{j \in W^{(u')}} p^{(u')}(j) - c(j) \quad \forall u' \in U \setminus u.$$

Among all vertices that are most preferable given set U , we select the one with the highest social welfare and set the α -parameters accordingly.

Theorem 7.3.6. *For the regulation that forbids the company to violate the constraints, the parameter vector α that maximizes the total social welfare, and the corresponding profit maximizing prices can be computed in polynomial time, given that the number of distinct item types m is constant.*

7.4 Computational results

In the summer of 2007, the European Commission decided to implement an EU-wide ceiling on the international roaming tariffs. In Table 7.1 we present the prices as they were, are and will be implemented (European Commission, Information Society, Media Directorate General 2007), where the inter-operator tariff is the maximum amount an operator may charge another operator for using the network. These are the costs for the operator for providing roaming service to the customers to call abroad.

	Summer 2007	Summer 2008	Summer 2009
Calls made abroad	0.5831	0.5474	0.5117
Calls received abroad	0.2856	0.2618	0.2261
Inter-operator tariff	0.3570	0.3332	0.3094

Table 7.1: Ceiling on prices in euro by the European Commission, including value added tax.

The goal of this practical study is to analyze the effect of properly chosen parameters on social welfare, and the advantage of using taxes instead of forbidding to violate the price regulation. We use data from a telephone operator that resemble reality. These data contain the phone usage of customers with a prepaid subscription during one month, March 2007, thus this data set was generated in a period before the introduction of the price regulation. For each customer, the data contain the number of minutes and times the mobile phone is used for different actions, for example, calling within the home country or abroad, sending a text message, etc. We determine for every customer which operator in the telephone market offers the cheapest possible total price for her complete contract. This price determines her *valuation* b_j . For this study, even though the data contains more information, we focus on optimizing the prices for roaming only, namely calling and receiving a call abroad. This is to say, we consider a problem in dimension 2, with prices p_1 for calling abroad, and p_2 for receiving a call abroad. We also impose the constraint that the price for receiving a call should not exceed the price for placing a call, that is, $p_1 \geq p_2$. If customer $j \in \mathcal{J}$ requests to call d_{j1} minutes from abroad and receives calls abroad for d_{j2} minutes, within a month, the price customer j has to pay is $p(j) = d_{1j}p_1 + d_{2j}p_2$.

7.4.1 Experiments

We apply the model and techniques from Section 7.3 to a data set containing 1366 customers. Also, we create one random sample out of this data set containing 500 customers. In the application of the first algorithm, we use the current price regulations imposed by the European Commission as described in Table 7.1. The costs for calling from abroad (the inter-operator tariff), c_1 , is also retrieved from the table, and the cost for the operator for a customer to receive a call is half of this. First, we forbid the operator to violate this price regulation by law (LAW), that is, a penalty $\tau_1 = \infty$. Second, we keep the price regulative parameters α_k as they are, but now we find a tax level $\tau_1 \in [0, 1]$ which maximizes social welfare (TAX). Note that there can be a range of feasible tax levels achieving the maximal social welfare by definition of LP2. This effect was also observed in our results. Third, we compute the optimal social welfare by optimizing over the price regulative parameters α_k (OPT). The computational study is conducted on a computer with an Intel Core2Duo CPU running at 1.86GHz and 2.0 GB of Ram.

Here, p_1 denotes the price per minute in euro for calling from abroad, and p_2 denotes the price per minute in euro for receiving a call abroad. The tables below give the complete results for all algorithms, that is,

- pricing strategy p_1 and p_2 ,
- regulative parameters α_1 , α_2 and τ_1 ,
- revenue $R = \sum_{j \in W} p(j)$,
- paid cost $C = \sum_{j \in W} c(j)$,
- amount subject to taxes $T = \sum_{j \in W} \sum_{k \in K} d_{jk} (p_k - \alpha_k)^+$,
- profit $\Pi = R - C$,
- total valuation $B = \sum_{j \in W} b_j$,
- total customer utility $U = \sum_{j \in W} \ln(b_j - p(j) + 1)$, and
- social welfare $S = U + \Pi$,

where for the tax level we present the computed lower and upper bound that lead to this result. Finally, we present the runtime in seconds. Tables 7.2, 7.3 and 7.4 display the results for the regulation as implemented in 2007, 2008 and 2009, respectively.

Complete data set				Sample			
	LAW	TAX	OPT		LAW	TAX	OPT
p_1	0.5830	0.7448	0.7448	p_1	0.5830	1.2545	1.2309
p_2	0.2856	0.5459	0.5459	p_2	0.2856	0.4049	0.1946
α_1	0.5831	0.5831	0.7448	α_1	0.5831	0.5831	1.2324
α_2	0.2856	0.2856	0.5492	α_2	0.2856	0.2856	0.3020
τ_1	∞	12.42%	∞	τ_1	∞	4.35%	∞
	∞	61.79%	∞		∞	71.69%	∞
R	3866.09	4101.72	4101.72	R	758.02	969.83	981.00
C	2370.82	1917.16	1917.16	C	464.95	283.91	308.25
T	0.00	973.87	0.00	T	0.00	506.86	0.00
Π	1495.27	2184.56	2184.56	Π	293.07	685.92	672.74
B	8248.06	7422.67	7422.67	B	2096.46	1668.07	1726.61
U	1338.93	1131.21	1131.21	U	585.62	344.71	365.92
S	2834.20	3315.77	3315.77	S	878.69	1030.63	1038.66
sec	133	125	125	sec	1	1	1

Table 7.2: Results for regulation in 2007.

Complete data set				Sample			
	LAW	TAX	OPT		LAW	TAX	OPT
p_1	0.5474	0.7448	0.7448	p_1	0.5474	1.2545	1.2309
p_2	0.2618	0.5459	0.5459	p_2	0.2317	0.2397	0.1946
α_1	0.5474	0.5474	0.7448	α_1	0.5474	0.5474	1.2324
α_2	0.2618	0.2618	0.5492	α_2	0.2618	0.2618	0.3020
τ_1	∞	3.78%	∞	τ_1	∞	52.40%	∞
	∞	61.48%	∞		∞	72.51%	∞
R	3650.63	4101.72	4101.72	R	742.33	978.15	981.00
C	2228.86	1789.35	1789.35	C	458.79	277.30	287.70
T	0.00	1169.08	0.00	T	0.00	528.36	0.00
Π	1421.77	2312.37	2312.37	Π	283.54	700.85	693.29
B	8275.30	7422.67	7422.67	B	2136.18	1698.15	1726.61
U	1375.74	1131.21	1131.21	U	603.60	354.08	365.92
S	2797.51	3443.59	3443.59	S	887.15	1054.92	1059.21
sec	136	136	134	sec	2	1	1

Table 7.3: Results for regulation in 2008.

Complete data set				Sample			
	LAW	TAX	OPT		LAW	TAX	OPT
p_1	0.5117	0.7448	0.7448	p_1	0.5117	1.2545	1.2309
p_2	0.2261	0.5459	0.5459	p_2	0.2213	0.2397	0.1946
α_1	0.5117	0.5117	1.5004	α_1	0.5117	0.5117	1.2324
α_2	0.2261	0.2261	1.4854	α_2	0.2261	0.2261	0.3020
τ_1	∞	0.00%	∞	τ_1	∞	33.67%	∞
	∞	61.13%	∞		∞	73.83%	∞
R	3567.25	4101.72	4101.72	R	714.42	978.15	981.00
C	2176.63	1661.54	1661.54	C	437.80	257.50	267.15
T	0.00	1371.31	0.00	T	0.00	557.35	0.00
Π	1390.62	2440.19	2440.19	Π	276.62	720.65	713.84
B	8456.72	7422.67	7422.67	B	2155.85	1698.15	1726.61
U	1418.53	1131.21	1131.21	U	618.39	354.08	365.92
S	2809.15	3571.40	3571.40	S	895.02	1074.73	1079.76
sec	133	133	306	sec	1	1	1

Table 7.4: Results for regulation in 2009.

We summarize our conclusions regarding the computational study from these results as follows. Introducing a tax (TAX) instead of enforcing the price regulation by law (LAW) leads to an increase in the social welfare. This suggests that a more liberal price regulative policy might have the potential to improve social welfare. The tax levels that are obtained using the algorithm are not trivial. In extreme cases, not observed here though, a company might be able to participate in the market where it would not be profitable to do so if violation of price regulations was forbidden. Not surprisingly, the social welfare is maximal when the α -parameters are optimized (OPT). In the sample, the social welfare is strictly larger in the latter case. For the complete data set, both algorithms yield the same social welfare. Concluding, it seems that the current EU practice does not yield the optimum, at least not with the data set and experimental setup that we use here.

7.5 Conclusion

First, we think it is an interesting result in its own that the given three-level program can indeed be solved in polynomial time by making use of linear programming techniques. Even though techniques are comparably simple

and crucially use the fact that the dimension m is constant, we believe it is not straightforward to come up with a polynomial time algorithm.

Second, on the more economic side, our computational results suggest that a more liberal price regulative policy, namely taxation instead of regulation by law, might lead to an increase in social welfare. But of course, this conclusion cannot be made hard as it depends very much on the choice of the social welfare function.

Bibliography

- Abrams, Z. (2006). Revenue maximization when bidders have budgets. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1074–1082. ACM-SIAM.
- Aggarwal, G., T. Feder, R. Motwani, and A. Zhu (2004). Algorithms for multi-product pricing. In J. Díaz, J. Karhumäki, A. Lepistö, and D. Sannella (Eds.), *Automata, Languages and Programming - ICALP 2004*, Volume 3142 of *Lecture Notes in Computer Science*, pp. 72–83. Springer.
- Aggarwal, G. and J. D. Hartline (2006). Knapsack auctions. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1083–1092. ACM-SIAM.
- Ahuja, R. K., T. L. Magnanti, and J. B. Orlin (1993). *Network flows*. New Jersey: Prentice Hall.
- Archer, A., C. Papadimitriou, K. Talwar, and É. Tardos (2003). An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 205–214. ACM-SIAM.
- Balcan, M. F. and A. Blum (2006). Approximation algorithms and online mechanisms for item pricing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pp. 29–35. ACM.
- Balcan, M. F., A. Blum, J. D. Hartline, and Y. Mansour (2005). Mechanism design via machine learning. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 605–614.

- Balcan, M. F., A. Blum, J. D. Hartline, and Y. Mansour (2008). Reducing mechanism design to algorithm design via machine learning. *Journal of Computer and System Sciences*, to appear.
- Balcan, M. F., A. Blum, and Y. Mansour (2007, July). Improved guarantees for revenue maximization in combinatorial auctions. Working paper.
- Baz, J. and G. Chacko (2004). *Financial derivatives: pricing, applications, and mathematics*. Cambridge University Press.
- Bazaraa, M. S., H. D. Sherali, and C. M. Shetty (1993). *Nonlinear programming: theory and algorithms* (2 ed.). John Wiley & Sons, Inc.
- Berger, A., A. Grigoriev, and J. van Loon (2008). Price strategy implementation. METEOR Research Memorandum RM/08/035, Universiteit Maastricht.
- Bernoulli, D. (1738). Specimen theoriae novae de mensura sortis (Translated from Latin into English by L. Sommer in 1954). In *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, Volume V, pp. 175–192.
- Bodlaender, H. and E. Penninx (2005, May). An elegant NP-completeness proof for profit maximization on paths. Personal communication.
- Bouthou, M., A. Grigoriev, S. van Hoesel, A. F. van der Kraaij, F. Spieksma, and M. Uetz (2007). Pricing bridges to cross a river. *Naval Research Logistics* 54(4), 411–420.
- Brandstädt, A., V. B. Le, and J. P. Spinrad (1999). *Graph classes: a survey*. Philadelphia: SIAM.
- Briest, P. (2007). *Computational aspects of combinatorial pricing problems*. Ph. D. thesis, Universität Dortmund.
- Briest, P. (2008). Uniform budgets and the envy-free pricing problem. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 808–819.
- Briest, P., M. Hoefler, and P. Krysta (2008). Stackelberg network pricing games. In S. Albers and P. Weil (Eds.), *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science*, pp. 133–142.

- Briest, P. and P. Krysta (2006). Single-minded unlimited supply pricing on sparse instances. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1093–1102. ACM-SIAM.
- Briest, P. and P. Krysta (2007). Buying cheap is expensive: hardness of non-parametric multi-product pricing. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 716–725. ACM-SIAM.
- Brotcorne, L., M. Labbé, P. Marcotte, and G. Savard (2001). A bilevel model for toll optimization on a multicommodity transportation network. *Transportation science* 35(4), 345–358.
- Chawla, S., J. D. Hartline, and R. Kleinberg (2007). Algorithmic pricing via virtual valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*, pp. 243–251.
- Courcoubetis, C. and R. Weber (2003, March). *Pricing communication networks – economics, technology and modelling*. New York: Wiley.
- Cramton, P., Y. Shoham, and R. Steinberg (2006). *Combinatorial auctions*. MIT Press.
- Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton, New Jersey: Princeton University Press.
- Demaine, E. D., U. Feige, M. Hajiaghayi, and M. R. Salavatipour (2006). Combination can be hard: approximability of the unique coverage problem. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 162–171. ACM-SIAM.
- Dewez, S. (2004). *On the toll setting problem*. Ph. D. thesis, Université Libre de Bruxelles.
- Edmonds, J. (1965). Matching and a polyhedron with 0-1 vertices. *Journal of Research of the National Bureau of Standards, B* 69, 125–130.
- Elbassioni, K., R. Sitters, and Y. Zhang (2007). A quasi-PTAS for profit-maximizing pricing on line graphs. In L. Arge and E. Welzl (Eds.), *Proceedings of the 15th Annual European Symposium on Algorithms*, Volume 4698 of *Lecture Notes in Computer Science*, pp. 451–462. Springer.
- European Commission, Information Society, Media Directorate General (2007, June). EU roaming regulation: using your mobile abroad in the EU is now much cheaper (factsheet 59).

- Foley, D. (1967). Resource allocation and the public sector. *Yale Economics Essays* 7, 45–98.
- Fulkerson, D. R. and O. A. Gross (1965). Incidence matrices and interval graphs. *Pacific Journal of Mathematics* 15, 835–855.
- Garey, M. R. and D. S. Johnson (1979). *Computers and intractability: a guide to the theory of NP-completeness*. New York: W. H. Freeman.
- Goldberg, A. V. and J. D. Hartline (2001). Competitive auctions for multiple digital goods. In F. M. auf der Heide (Ed.), *Algorithms - ESA 2001*, Volume 2161 of *Lecture Notes in Computer Science*, pp. 416–427. Springer.
- Goldberg, A. V., J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright (2006). Competitive auctions. *Games and Economic Behavior* 55, 242–269.
- Golumbic, M. C. and P. L. Hammer (1988). Stability in circular arc graphs. *Journal of Algorithms* 9, 56–63.
- Golumbic, M. C. and R. E. Jamison (1985). The edge intersection graphs of paths in a tree. *Journal of Combinatorial Theory, Series B* 38, 8–22.
- Grigoriev, A., J. van Loon, R. Sitters, and M. Uetz (2006). How to sell a graph: guidelines for graph retailers. In F. Fomin (Ed.), *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, Volume 4271 of *Lecture Notes in Computer Science*, pp. 125–136. Springer.
- Grigoriev, A., J. van Loon, R. Sitters, and M. Uetz (2009). Optimal pricing of capacitated networks. *Networks* 53(1), 79–87.
- Grigoriev, A., J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld (2007). Bundle pricing with comparable items. In L. Arge, M. Hoffmann, and E. Welzl (Eds.), *Algorithms - ESA 2007*, Volume 4698 of *Lecture Notes in Computer Science*, pp. 475–486. Springer.
- Grigoriev, A., J. van Loon, M. Sviridenko, M. Uetz, and T. Vredeveld (2008). Optimal bundle pricing with monotonicity constraint. *Operations Research Letters* 36, 609–614.
- Grigoriev, A., J. van Loon, and M. Uetz (2008a). Algorithms for optimal price regulations. In C. Papadimitriou and S. Zhang (Eds.), *Proceedings of the 4th International Workshop On Internet And Network Eco-*

- nomics (WINE)*, Volume 5385 of *Lecture Notes in Computer Science*, pp. 362–373. Springer.
- Grigoriev, A., J. van Loon, and M. Uetz (2008b). On the complexity of the highway pricing problem. METEOR Research Memorandum RM/08/030, Universiteit Maastricht.
- Gul, F. and E. Stacchetti (1999). Walrasian equilibrium with gross substitutes. *Journal of Economic Theory* 87, 95–124.
- Guruswami, V., J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry (2005). On profit-maximizing envy-free pricing. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1164–1173. ACM-SIAM.
- Hartline, J. D. and A. R. Karlin (2007). *Algorithmic game theory*, Chapter 13 - Profit maximization in mechanism design, pp. 331–361. Cambridge University Press.
- Hartline, J. D. and V. Koltun (2005). Near-optimal pricing in near-linear time. In F. K. H. A. Dehne, A. López-Ortiz, and J.-R. Sack (Eds.), *Algorithms and Data Structures - WADS 2005*, Volume 3608 of *Lecture Notes in Computer Sciences*, pp. 422–431. Springer.
- Heilporn, G., M. Labbé, P. Marcotte, and G. Savard (2007). A polyhedral study of the network pricing problem with connected toll arcs. Technical report, Graphes et Optimisation Mathématique (GOM), Université Libre de Bruxelles.
- Hoffman, A. and J. Kruskal (1956). *Linear inequalities and related systems*, pp. 223–246. Princeton University Press.
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. Master’s thesis, University of Chicago, Chicago, Illinois.
- Kuhn, H. W. and A. W. Tucker (1951). Nonlinear programming. In J. Neyman (Ed.), *Proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, California, pp. 481–492. University of California Press.
- Labbé, M., P. Marcotte, and G. Savard (1998). A bilevel model of taxation and its application to optimal highway pricing. *Management science* 44, 1608–1622.

- Lehmann, D., L. I. O’Callaghan, and Y. Shoham (2002, September). Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM* 49(5), 1–26.
- Marcotte, P. and G. Savard (2005). *Graph theory and combinatorial optimization*, Chapter 7 - Bilevel programming: a combinatorial perspective, pp. 191–218. Springer.
- Möhring, R. H. (1985). Algorithmic aspects of comparability graphs and interval graphs. In I. Rival (Ed.), *Graphs and Order*, pp. 41–101. Dordrecht: Reidel.
- Mu’alem, A. and N. Nisan (2002). Truthful approximation mechanisms for restricted combinatorial auctions. In *AAAI/IAAI*, pp. 379–384.
- Myerson, R. (1981). Optimal auction design. *Mathematics of Operations Research* 6, 58–73.
- Nemhauser, G. L. and L. A. Wolsey (1988). *Integer and combinatorial optimization*. New York: John Wiley & Sons.
- Nisan, N. and A. Ronen (1999). Algorithmic mechanism design. In *Proceedings of the 31st STOC*, pp. 129–140.
- Papadimitriou, C. H. and K. Steiglitz (1998). *Combinatorial optimization: algorithms and complexity*. New York: Dover Publications.
- Rendleman, J. (2001, August). Customer data means money. Information-Week.
- Roch, S., G. Savard, and P. Marcotte (2005). An approximation algorithm for Stackelberg network pricing. *Networks* 46(1), 57–56.
- Rothkopf, M. H., A. Pekec, and R. M. Harstad (1998). Computationally manageable combinatorial auctions. *Management Science* 44(8), 1131–1147.
- Rusmevichientong, P., B. van Roy, and P. W. Glynn (2006). A nonparametric approach to multi-product pricing. *Operations Research* 54(1), 82–98.
- Rusmevichientong, P. (2003). *A non-parametric approach to multi-product pricing: theory and application*. Ph. D. thesis, Stanford University.
- Savage, L. J. (1972). *The foundations of statistics* (2 ed.). Courier Dover Publications.

- Shavel, S. (2004). *Foundations of economic analysis of law*. Harvard University Press.
- Tarjan, R. E. (1985). Decomposition by clique separators. *Discrete Mathematics* 55, 221–232.
- van der Kraaij, A. F. (2004). *Pricing in networks*. Ph. D. thesis, Universiteit Maastricht.
- van Hoesel, S. (2006). An overview of Stackelberg pricing in networks. METEOR Research Memorandum RM/06/042, Universiteit Maastricht.
- Vanderbei, R. J. (2001). *Linear programming: foundations and extensions*. Springer.
- Vicente, L. N. and P. H. Calamai (1994). Bilevel and multilevel programming: a bibliography review. *Journal of Global Optimization* 5, 291–306.
- von Stackelberg, H. F. (1934). *Marktform und gleichgewicht*. Vienna: Verlag von Julius Springer.
- Walras, L. (1954). *Elements of pure economics*. London: Allen and Unwin.
- Zuckerman, D. (2006). Linear degree extractors and the inapproximability of MAX CLIQUE and CHROMATIC NUMBER. In *Proceedings of the 38th annual ACM Symposium on Theory of Computing*, pp. 681–690. ACM-SIAM.

Index

A

Affine function ..18, 87, 88, 106, 121
Affine pricing 16, 21, 88, 121
 Algorithm 90
Algorithmic pricing 11, 12
Approximation algorithm 43, 54, 55,
 69, 111
APX 19
Arrangement of linear
 inequalities 89, 97
 Edge 97
 Vertex .. 89, 93, 95, 97, 124, 132
Average valuation 48

B

Bilevel program 20, 122, 129
Bundle 13, 27, 48, 66

C

Circular arc graph 79
Consecutive ones property 68
Contract 17, 87, 106, 121

D

Demand 88, 106, 120
Demand set 96
Dynamic program 70, 71, 110

E

Edge *see* Arrangement of linear
 inequalities
Envy-free 14, 28, 66, 89, 121
EPT graph 80
EqualCardinalitySubsetSum 36

F

Fully polynomial time approximation
 scheme (FPTAS) 20, 74

G

Gadget 37, 59

H

Highway problem 16, 35, 58, 67

I

IndependentSet 29, 50, 81, 94
Index set 97, 132
Inhomogeneity 48
Interval graph 78

L

Laminar family 20, 67
Line graph 80
Local search 96, 111

M

Mechanism design 12

Index

M
Monotonicity
 Average valuations 49
 Constraint 28
 Valuations 48

N
Near-optimal solution 93

P
Partition 59
Polynomial time approximation
 scheme (PTAS) .. 19, 33, 34
 Fully (FPTAS) 20, 74
 Quasi (QPTAS) 20

Q
Quasi-polynomial time approxima-
 tion scheme (QPTAS) 20

R
Regulation 18, 120
Risk-averse 130
Risk-neutral 130
Roaming 18, 120

S
Simplex algorithm 97
Single item pricing 13, 27, 47, 65
Single-minded 13, 27, 48, 66, 88,
 106, 120
Social welfare 120, 130
Stackelberg game 11
Straight line algorithm 109
SubsetSum 36
Supply
 Limited 14, 71, 92
 Unlimited 13, 27, 47, 69, 89,
 106, 121

T
Target market 106

Three-level program 122
Tollbooth problem 15, 65
Totally unimodular 68

U
Unit-demand 13, 95
Utility 13, 95, 130
 Logarithmic function 130
 Transferable 131

V
Valuation 13, 17, 27, 48, 66, 88, 106,
 121
Vertex ... *see* Arrangement of linear
 inequalities

W
Winner 27, 48, 66, 89, 106, 121

Nederlandse Samenvatting

Voor ieder bedrijf is het bepalen van prijzen een zeer belangrijke, maar ook gecompliceerde taak. Het doel is immers om zoveel mogelijk opbrengst te genereren. Wanneer beslissingen over prijzen worden genomen zonder onderzoek en analyse kan een bedrijf een groot deel van de opbrengst mislopen. Een te lage prijs kan voor verlies zorgen als geïnteresseerde consumenten meer hadden willen betalen voor het product. Maar, het zorgt er wel voor dat er meer consumenten worden aangetrokken die met een hogere prijs het product misschien niet zouden aanschaffen. Een te hoge prijs levert meer op per verkocht product, maar wellicht kopen sommige consumenten het product niet omdat het simpelweg te duur is. Kortom, een goede marktkennis en algoritmische analyse zijn van wezenlijk belang bij het bepalen van de juiste prijzen om de opbrengst te maximaliseren.

Het prijsprobleem is een spel tussen het bedrijf en de consumenten. Eerst bepaalt het bedrijf de prijzen voor de producten. Vervolgens, deze prijzen beschouwend, kiezen de consumenten welke producten ze willen kopen en of ze bereid zijn om de gezette prijzen te betalen. Dit spel op twee niveaus, waar het bedrijf de leider is en de consumenten de volgers zijn, is een instantie van een Stackelberg spel.

Als leider van dit spel moet het bedrijf de eerste stap zetten. Als het bedrijf al toegang heeft tot informatie over de voorkeuren van de consumenten op het moment dat de prijzen bepaald moeten worden, is dat vanzelfsprekend een groot voordeel. Het verkrijgen van fundamentele kennis over de doelgroep is tegenwoordig makkelijker dan ooit tevoren, voornamelijk vanwege het gebruik van internet. Een bedrijf kan gegevens kopen over bijna elk product, iedere industrie, trend en interesse, op een manier die

het beste past bij de specifieke behoeften van het bedrijf. Vanwege dit grote aanbod en het gemak om informatie te verspreiden, weten bedrijven wat consumenten kopen, hoeveel ze uitgeven en hoe vaak een product wordt gekocht. Die informatie kan veel geld waard zijn als ze ten volste wordt gebruikt. *Algoritmische prijsbepaling* is de studie waarin het wiskundige probleem wordt bekeken dat een bedrijf ondervindt tijdens het bepalen van de prijzen met als doel de opbrengst te maximaliseren, gebruik makend van alle informatie over de voorkeuren van de consumenten. Bovendien is het algoritmische prijsprobleem niet alleen van belang in een situatie waarin veel informatie beschikbaar is. Ook wanneer de voorkeur van een consument persoonlijke informatie is, en dus niet vooraf bepaald kan worden door het bedrijf, is het zeer belangrijk om een goed algoritmisch begrip te hebben over hoe de prijzen de opbrengst beïnvloeden.

In dit proefschrift worden twee algoritmische prijsproblemen bekeken, namelijk het *productsgewijze* prijsprobleem en het *affiene* prijsprobleem.

In het productsgewijze prijsprobleem beschikt het bedrijf over een verzameling producten en voor elk individueel product moet de prijs worden bepaald. Elke consument vraagt een of meerdere deelverzamelingen van producten, die *bundels* worden genoemd. Elke bundel bevat hoogstens één kopie van elk product. Voor iedere bundel die een consument vraagt, heeft ze een *waardebepaling*. Dit is het maximale bedrag dat ze voor die specifieke bundel wil betalen. De totale prijs van een bundel is gelijk aan de som van de prijzen van alle producten in de bundel. Een consument koopt de producten in een gekozen bundel alleen als de prijs niet hoger is dan haar waardebepaling. Het doel van het bedrijf is om de opbrengst te maximaliseren, waarbij de opbrengst gelijk is aan de totale som van prijzen van alle verkochte producten.

Een specifieke instantie van het productsgewijze prijsprobleem is het zogenaamde *tolhuisjes* probleem. Dit probleem vindt zijn oorsprong in het bepalen van tolprijzen op delen van een snelwegennetwerk. Consumenten reizen over verscheidene, opeenvolgende delen van het netwerk. De waardebepaling wordt berekend als zijnde de kosten van het reizen over alternatieve wegen. Het doel van de eigenaar van het wegennetwerk is om de opbrengst te maximaliseren. Meer specifiek is het *snelweg* probleem, waar de tolprijzen worden bepaald op delen van één enkele snelweg.

Het algemene productsgewijze prijsprobleem en het snelweg probleem worden beschreven in het eerste deel van dit proefschrift. In Hoofdstukken 2 en 3 worden deze problemen bekeken onder verschillende natuurlijke mono-

toniciteitsrestricties. In Hoofdstuk 4 is het aanbod van producten beperkt. Hierdoor wordt het bepalen van een allocatie van producten aan consumenten een uitdaging.

In het affiene prijsprobleem bepaalt het bedrijf de prijs van verscheidene producttypes. Een producttype kan tastbaar of digitaal zijn, maar ook een dienst, transport, het starten van een proces of het aangaan van een abonnement. Elke consument vraagt een *contract* dat is gebaseerd op haar persoonlijke wensen betreffende de beschikbare producttypes. Dit contract mag meerdere kopieën van een type bevatten. Een consument accepteert het contract als de prijs niet hoger is dan haar waardebeoordeling. Deze waardebeoordeling kan worden gezien als het bedrag dat een consument voor hetzelfde contract moet betalen bij een concurrerend bedrijf. Hierdoor koopt de consument het contract als het het goedkoopste is in de markt. Mocht het bedrijf in staat zijn om een andere waardebeoordeling te verkrijgen, kan deze natuurlijk worden gebruikt. De prijs van een contract is gelijk aan een constante prijs per producttype voor elke eenheid die een consument vraagt, en mogelijk een vast bedrag voor het accepteren van het contract. Dit bedrag is afhankelijk van de consument zelf. Anders gezegd, gegeven de prijzen bepaald door het bedrijf, de prijs die een consument moet betalen voor haar contract is een *affiene functie* van haar persoonlijke vraag.

Dit probleem wordt beschreven in Hoofdstuk 5. Vervolgens, in Hoofdstuk 6, wordt het probleem bekeken dat een bedrijf tegenkomt bij het implementeren van nieuwe prijzen. Het is namelijk niet altijd mogelijk om dit in maar één tijdstap te doen vanwege de bemoeienis van autoriteiten, de strategie ten opzichte van concurrenten of de perceptie van consumenten. Dit alles kan het bedrijf dwingen om de nieuwe prijzen geleidelijk te introduceren. In Hoofdstuk 7 wordt de huidige prijsregulatie op roamingtarieven bekeken. Roaming is het gebruik maken van een mobiele telefoon buiten de geografische dekking van het thuisnetwerk. Deze regulatie is in de zomer van 2007 ingesteld door de Europese Commissie. Het doel van de regulatie is om eerlijkere prijzen te verkrijgen en om de markt transparanter te maken. Naast het bekijken van de huidige situatie, worden ook alternatieve regulaties aangedragen en oplossingsmethoden beschreven.

Curriculum Vitae

Joyce Maria Wilhelmina van Loon was born on June 30, 1982 in Heerlen, The Netherlands. In 2000, she received her Gymnasium diploma from the Sintermeertencollege in Heerlen. In September of that same year she started studying Econometrics with specialty Operations Research at Maastricht University. She received her Master's degree in August 2004. The Master thesis was written during an internship at the Research Institute for Knowledge Systems in Maastricht. From September 2004 till March 2005, Joyce was employed as a research and teaching assistant at the same university. In April 2005, she started as a PhD student in the Quantitative Economics department. The results of her research are presented in this thesis.