TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Pricing in Non-Convex Markets

Dissertation

Richard Littmann

Fakultät für Informatik

Pricing in Non-Convex Markets

Richard Littmann

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender: Prof. Dr. Florian Matthes

Prüfende/-r der Dissertation:

1. Prof. Dr. Martin Bichler
2. Prof. Dr. Peter Gritzmann

Die Dissertation wurde am 29.01.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 09.03.2021 angenommen.

Abstract

The Vickrey-Clarke-Groves (VCG) mechanism is known to be the only efficient, strategyproof mechanism when agents have quasi-linear utilities. However, it is not applicable outside of quasi-linear environments and depends on optimal allocations which are rarely available in non-convex markets. We present three projects motivated by real-world applications where the assumptions of the VCG mechanism are not given. We use prices to achieve practically important properties such as strategyproofness or core-stability.

In the absence of optimal allocations, the VCG mechanism loses its strong game-theoretical guarantees. We consider a network design problem in the telecommunications sector where the underlying non-convex allocation problem is NP-hard and thus needs to be approximated in practice. We extend approximation algorithms for the underlying Steiner minimum tree problem into strategyproof mechanisms and consider three variants of the deferred-acceptance auction. Results of a computational study show that the resulting strategyproof mechanisms achieve high average efficiency.

Moreover, the VCG mechanism cannot be applied in markets where utilities are non-quasi-linear. An application in secondary markets for airport time slots constitutes the second project. The market under consideration is non-quasi-linear due to budget constraints airlines may have in this cost intensive industry. No strategyproof mechanism can be obtained, but core-stable solutions can be computed if the core is non-empty. We use a mixed-integer bilevel linear program leveraging problem-specific constraints and solve problem instances of relevant size despite the inherent complexity (Σ_2^{P}-hard). Additionally, ignoring budget constraints does not only lead to a loss in welfare, but also to unstable solutions, arguing in favor of markets beyond quasi-linear utilities.

Furthermore, the VCG mechanisms is not applicable in online settings where bids arrive over time. Real-time congestion pricing serves as a practical example. Currently, most roads are not priced at all, and even if road pricing mechanisms are in place, they exclusively use non-responsive pricing schemes. We implement a dynamic pricing scheme by using an online linear programming algorithm. A computational study shows that our mechanism leads to a reduction in congestion when compared against low or high static, area-based prices and unpriced roads.

Despite the computational hardness of the underlying non-convex allocation problems, we show that practically relevant problem sizes can be solved for all problems. These results provide insights into markets for which the VCG mechanism is not applicable due to unavailability of optimal allocations or non-quasi-linear utility functions.

Acknowledgments

I am very grateful for all the assistance I received by Martin Bichler and Stefan Waldherr. Thank you for your ideas, your patience, and the opportunity to work with you. Thank you to all those I have worked with both at the chair and at AdONE for making my studies a very enjoyable experience. I would not have chosen this field if it was not for Michael Prautzsch, who first sparked my interest in informatics. I would like to thank my parents for their continuous support, my brother for following the family tradition, and my aunt and grandmother for their interest in my work. Finally, I am deeply indebted to my wife Maria, who makes me a better person.

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) (GRK 2201).

Contents

1 Introduction

In a society where professions are highly specialized, trade plays a crucial role in the distribution of goods and services. Beyond bilateral trade, there is a need for markets which enable the emergence of efficient such distributions. However, the design of those markets has many inherent challenges as buyers and sellers usually pursue opposite goals. Incomplete information about their negotiation partners is the central challenge for market participants as parties keep their true valuations secret in order to come to a more favorable agreement. Prices can be used to communicate (and thus also elicit) valuations in a market. Whenever the seller suggests a price, the buyer can either accept or reject a trade at this price. By accepting a price, the buyer implicitly communicates that the item is worth at least the price and, analogously, can indicate that the item is worth less than the price by rejecting the offer. Despite its importance, the communication aspect is but one facet of the benefits prices offer. Choosing correct prices can also lead to a state of equilibrium in the market, discourage strategic manipulation by rational market participants, or maximize revenue of the seller.

The foundations for the corresponding theoretical concepts lie in *Mechanism Design Theory*, a field concerned with the development of economic mechanisms for rational bidders with private information. In 2007, the Nobel Memorial Prize in Economics was awarded to Leonid Hurwicz, Eric S. Maskin, and Roger B. Myerson for their contributions to this field of economic theory (Mookherjee, 2008). Based on axiomatic assumptions, one can derive mechanisms that satisfy relevant properties in regard to strategic behavior or perceived fairness (Börgers and Krahmer, 2015). In economic contexts, mechanisms with monetary transfer are more commonly used as they allow to transfer utility by means of payments. In this dissertation, we focus on mechanisms with monetary transfers in economic settings with special attention to setting prices (pricing mechanisms).

In general, such mechanisms can be decomposed into two functional units: allocation and pricing. In the former, the mechanism determines the distribution of items to market participants. Social welfare maximization is the most frequently optimized objective here, but others, such as maximizing the number of allocated items, are possible. Prices are then used to offset the utility which was transferred via items. Carefully designed pricing rules can set strong incentives for truthful behavior or enable equilibrium states where no coalition of participants wants to deviate.

Due to the inherent interdisciplinary nature of pricing problems, researchers from many different fields, including economics, mathematics, operations research, and computer science, contributed to the existing literature in significant ways. At this intersection, auctions constitute an important class of pricing mechanisms. While Paul R. Milgrom and Robert B. Wilson just recently received the Nobel Memorial Prize in Economics for their work in this field, auctions date back 2500 years. Auctions have come a long way since ancient Babylon and experience an increase in popularity in recent times. Reasons for this trend can be found in improved auction formats incorporating real-world requirements to a sufficient degree and larger computational power allowing to solve allocation problems that were formerly considered intractable. Nowadays, auctions are used to sell off perishable goods (e.g., flowers using a Dutch auction), one-of-a-kind items (e.g., a painting using an English Auction), and even public goods such as licenses for fishery or the utilization of specific bands of the electromagnetic spectrum. Online marketplaces offer private sellers a simple way to use an auction format for their sale, a popular solution due to the insufficient knowledge private sellers have of the market. Yet, suitable pricing mechanisms are still missing for many applications as a considerable gap between theory and practice remains. The Vickrey-Clarke-Groves (VCG) mechanism serves as a prominent example of an elegant theory rarely applicable in practice.

The celebrated VCG mechanism is a well-known example of a truthful mechanism (Clarke, 1971; Vickrey, 1961; Groves, 1973). Rested on an optimal allocation, it computes a bidder's payment based on the opportunity cost of this bidder's participation. Green and Laffont (1977) have shown that this is in fact the only efficient mechanism where no single bidder can gain by deviating from truthful bidding. Unfortunately, the assumptions in theory only seldomly translate to practice, and the VCG mechanism is rarely used in real-world applications (Rothkopf, 2007). This dissertation offers pricing mechanisms for markets where practical needs are not in line with central assumptions underlying the VCG mechanism.

The VCG mechanism assumes an allocation problem that can be solved to optimality. However, the allocation problem in many markets is computationally hard and thus does not allow for an optimal solution in practically feasible time. This is often the case when participants are interested in not only a single item but rather aim to trade certain bundles thereof (e.g., in spectrum auctions (Bichler and Goeree, 2017)). Problems of such combinatorial nature are notoriously difficult to solve due to the large number of possible combinations. Even if approximations are available, using approximate solutions in the VCG mechanism almost always leads to the loss of its game-theoretical properties (Lehmann et al., 2002)[1], enabling market participants to benefit from strategic manipulation. Markets where bids arrive in an online fashion, i.e., one by one over time, pose a similar challenge to the VCG mechanism. As decisions regarding the acceptance of bids need to be made at their arrival and the auctioneer lacks knowledge about subsequent bids, these online problems cannot be solved to optimality. Thus, there is a need for new practical mechanisms in these markets.

Secondly, the quasi-linear setting, which the VCG mechanism is designed for, has its inherent restrictions. In the quasi-linear setting, utility is transferable by monetary payments without limitation. In other words, theory assumes that market participants have sufficient capital to pay up to their valuation, while, in practice, many market participants may have private budget constraints. Markets where budget constraints need to be considered cannot be modeled using quasi-linear utility functions and thus require pricing mechanisms other than VCG.

To conclude, practice requires the development of new pricing mechanisms that relax assumptions of the VCG mechanism to achieve greater applicability. We present three such pricing mechanisms with applications in network procurement, airport time slot allocation, and congestion pricing. We use prices to achieve important mechanism design desiderata and provide brief descriptions of the projects in the following paragraphs. In our first project, we consider a computationally challenging problem in a network design market stemming from an application in the telecommunications industry. The auctioneer uses a reverse auction to buy network links from suppliers to establish connectivity between certain nodes in a network. While a variety of approximations is available, the VCG mechanism is not applicable since the allocation problem cannot be solved to optimality. Based on a result by Myerson (1981), we develop mechanisms based on approximate solutions and use prices to set incentives that inhibit beneficial strategic manipulation by individual market participants.

[1] Algorithms which are maximal-in-range (MIR) constitute an exception. However, approximations are almost never MIR by accident. `http://timroughgarden.org/w14/l/l28.pdf`, accessed: 28.11.2020

We consider a secondary market for airport time slots in our second project. Time slots are utilized at over-demanded airports and constitute the right to take-off/land at an airport at a certain time. Since, in the secondary market, airlines trade amongst themselves, we use an exchange rather than an auction format. The exchange generalizes the auction setting in that it has multiple buying and selling market participants, while an auction is restricted to a one-to-many relationship. Additionally, the airline industry is financially taxing, and many airlines face budget constraints. Due to the budget constraints, the problem cannot be modeled using quasi-linear utility functions, disqualifying the VCG mechanism. We suggest an approach where airlines are allowed to state both their valuations and their budget, making the problem harder but more accurately reflecting the needs in practice. Prices need to be chosen carefully in order to find a combination of allocation and prices that is stable even against deviation by groups of participants. To this end, we use a bilevel formulation of the market and leverage efficient problem-specific constraints to compute solutions for this computationally challenging, Σ_2^{P}-hard, problem.

Similar to airport time slots, road capacity is a scarce resource in major cities. Allocating it efficiently constitutes our third project. Road capacity is often not priced at all, leading to congestion with severe consequences for the environment (Chin, 1996), health (Currie and Walker, 2011; Simeonova et al., 2018), and the economy[2]. Pricing mechanisms currently implemented in practice are based on either entering an area or traveled distance, not on demand, but there is reason to believe that prices which respond to the current traffic situation result in a more meaningful reduction of congestion than static prices (Cheng et al., 2017). This requires price updates to be computed almost instantaneously and on incomplete information as drivers request road capacity one by one over time. The VCG mechanism cannot be implemented since no optimal allocations are available. Instead, we propose a real-time pricing mechanism and use a fast online approximation to compute prices which provide guidance for drivers by raising road prices dependent on utilization.

[2]See, for example: `https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf`, accessed: 02.11.2020

Outline

The remainder of this thesis is organized as follows: In Chapter 2, notation, concepts, and methods important to the projects of this dissertation are briefly outlined. We introduce notation and definitions standard to non-convex markets as well as important mechanism design desiderata. Furthermore, we describe the application domains of the following chapters. Chapter 3 presents the first project on strategyproof mechanisms for network procurement. In Chapter 4, we present work on secondary airport time slot markets with financially constrained bidders. The last project on real-time congestion pricing is presented in Chapter 5. Finally, we discuss results and contributions in Chapter 6.

2 Scientific Context

In this chapter, we introduce terms which are prerequisites to Chapters 3, 4, and 5. However, this chapter cannot replace the stimulating read of an introductory textbook. To this end, we recommend Bichler (2017) and Krishna (2009). We start by describing standard notation and give desirable properties. Afterwards, we introduce markets beyond standard assumptions and corresponding solution techniques. We conclude by describing real-world applications and reviewing related literature. Topics are presented largely on their own as they constitute necessary fundamentals for the following chapters. We indicate connections to Chapters 3, 4, and 5 wherever appropriate. Notation and definitions mostly follow Bichler (2017).

2.1 Quasi-linear Markets

Auctions are mechanisms where the outcome is solely dependent on the bids stated by market participants. Since auctions are the predominant class of pricing mechanisms and our projects consider auctions and an exchange (a more general auction), we use notation and take assumptions standard in combinatorial auction theory (Blumrosen and Nisan, 2007). Throughout this chapter, we consider markets with n bidders, m indivisible items, and a single auctioneer s_0 (unless stated otherwise). Market participants are categorized as either buyers or sellers according to their intentions in the market. Unless stated otherwise, we will use $\mathcal{I}$ to refer to the set of bidders and assume buying bidders (and a selling auctioneer). Every bidder $i \in \mathcal{I}$ has a valuation $v_i(x) \in \mathbb{R}$ for every allocation of items $x = (S_1, \ldots, S_n)$. Here, x is the allocation computed by the market mechanism such that bidders $1, \ldots, n$ receive bundles $S_1, \ldots, S_n$ with S being a subset of the m indivisible items $\mathcal{K}$ ($\supseteq S$) and X is the set of allocations. In general, these valuations might be dependent on any circumstance affecting the bidder, including personal preference or estimated realizable profit, and are expressed in terms of monetary units. The valuation function for bidder i can be formally defined as $v_i : X \rightarrow \mathbb{R}$. We further assume bidders' valuations to solely depend on their own assessment, i.e., they are independent of whether a competitor gets a certain bundle. This class of utility functions is referred to as *independent private valuations*. In this setting, $v_i(x)$ is thus only dependent on the bundle S bidder i receives, and hence $v_i(x) = v_i(S)$. Moreover, we adopt the standard assumption of *quasi-linear utilities* to model the benefit u_i bidder i receives in outcome $o = (x, p)$. An outcome is defined by its allocation x and a price or payment vector $p \in \mathbb{R}^n$.

Definition 2.1: Quasi-linear utility function. *Under independent private valuations, a quasi-linear utility function $u : O \rightarrow \mathbb{R}$ of a bidder $i \in \mathcal{I}$ is given by*

$$u_i(o) = v_i(S) - p_i$$

where $o = (x, p)$ is an element in the outcomes O, $v_i : X \rightarrow \mathbb{R}$ is a valuation function, and $p_i \in \mathbb{R}$ refers to the ith element in payment vector $p \in \mathbb{R}^n$.

In other words, $p_i \in \mathbb{R}$ refers to the payment bidder i receives or is obliged to make. Quasi-linearity implicitly includes the assumption of *transferable utilities*, i.e., utilities can be transferred between participants without constraints via monetary payments. We further assume rational behavior, i.e., participants always try to maximize their utility value u_i. This definition can also be applied to sellers which receive payments and might incur a loss in utility by selling items. However, unless stated otherwise, we adopt the standard assumption of a seller's valuation function being 0 for all allocations $x \in X$ (i.e., a situation of sell-off).

Bidding Languages

Bidding in a market requires precise communication of bids. This is straightforward when only a single item is sold. However, there are many contexts where multiple items are auctioned off at the same time. Examples include, but are not limited to, spectrum auctions (Bichler et al., 2014), airport time slots (Avenali et al., 2015), the London bus routes market, and procurement of freight transportation (both discussed in Cramton et al. (2006)). If all bidders had additive valuations $(v_i(A) + v_i(B) = v_i(\{A, B\}), \forall i \in \mathcal{I}, \forall A, B \in \mathcal{K})$, items could be traded independently in a series of sequential or in parallel auctions. However, the whole may be worth more than the sum of its parts. For example, this is the case for airport time slots (rights to take-off/land) where a single slot is difficult to use, but a pair of interrelated slots can be very valuable (items are complements). The opposite is true when items are substitutes, i.e., a bidder is only interested in acquiring one of two (or several) possible items. Examples for this can be found in markets for natural gas or certain agricultural contracts (Milgrom and Strulovici, 2009). Bidding in a parallel or sequential series of single-item auctions puts bidders at risk of the *exposure problem*. This problem describes the phenomenon of bidders being left with either a subset of the desired, complementary items or several substitute items. Using multi-item formats with package bids, which allow bidders to specify combinations of items, can prevent the exposure problem but require an effective way for bidders to express their valuations. XOR and OR bidding languages are the most important bidding languages in combinatorial auctions as they are the most fundamental and serve as a basis for a multitude of other bidding languages.

In an XOR bidding language, bids are mutually exclusive and at most one bid can be accepted per bidder. While this allows the expression of valuations for every bundle, it requires stating an exponential number of bids. In general, an auction with m items requires $2^m - 1$ bids per bidder. Even in a small auction with 20 items, each bidder would need to state more than a million ($2^{20} - 1$) bids. In practice, this is infeasible and leads to the *missing bids problem* - essentially, the allocation problem is incomplete due to the inability of bidders to state all their bids (Bichler et al., 2020c).

In contrast, an OR bidding language implicitly assumes bidders are willing to receive any disjoint combination of their bids for the aggregate of the corresponding prices. This reduces the number of required bids, but can only express valuations without substitutabilities, such as additive and super-additive valuations (where the whole is weakly greater than the sum of its parts). Even in a simple auction with two items, a bidder who wants either one of those items cannot express their valuation function accurately using an OR bidding language. In general, every combination of bids can be accepted as long as bids do not overlap in a way that would make accepting all of them infeasible due to insufficient supply. Dummy items can be used to create an overlap of artificial items. This extension known as OR* has the same expressiveness as the XOR bidding language, uses the same number of bids, and requires a number of dummy items that is quadratic in the number of bids (Nisan, 2006).

Several additional combinations of OR and XOR have been proposed to adhere to particular market needs (Nisan, 2006). Moreover, specific bidding languages have been designed for certain applications, e.g., for 5G auctions (Bichler et al., 2020c), to express characteristics of a supplier's cost functions in markets of scope and scale (Bichler et al., 2011), for online advertising (Lahaie et al., 2013), and for combinatorial exchanges (Lubin et al., 2008). In this line, we suggest a bidding language that combines elements of both XOR and OR for the secondary airport time slots market in our second project (Chapter 4). For a thorough introduction to bidding languages, we refer the interested reader to Nisan (2006).

2.2 Desiderata, Payment Rules and Incentives

While certain scenarios require dedicated performance indicators, some properties are generally considered desirable for pricing mechanisms. Let $\mathcal{M}(f, p)$ be a pricing mechanism with $f : B \to X$ and $p : B \times X \to \mathbb{R}^n$ being the allocation and payment function, and B referring to the set of all possible bid vectors $b \in B$. Note that bids are not necessarily truthful, i.e., there might be lying bidders i with $b_i(S) \neq v_i(S)$ for bundles $S \subseteq \mathcal{K}$. *Allocative efficiency* (sometimes short: efficiency) of a mechanism $\mathcal{M}$ relates the social welfare $\sum_{i \in \mathcal{I}} v_i(x)$ generated by the allocation x computed by f to the social welfare of an optimal, i.e., welfare-maximizing, allocation.

Definition 2.2: Allocative Efficiency. *Allocative efficiency is defined as*

$$\frac{f(b)}{OPT(b)^*}$$

where $f(b)$ refers to the welfare computed by an allocation function $f : B \to X$ and $OPT(b)^*$ refers to an welfare-maximizing allocation for a bid vector $b \in B$.

Mechanisms which maximize the social welfare $f(b) \in \arg\max_{x \in X} \sum_{i \in \mathcal{I}} v_i(x)$ are referred to as *efficient*. Finding a welfare-maximizing allocation is NP-hard for most combinatorial allocation problems. For problems in the complexity class of NP, only exponential time algorithms are known, and unless P=NP (an open Millennium Prize Problem), no polynomial time algorithms exist for this class of problems. Exponential time algorithms are improper for many practical applications as the runtime increases exponentially on input length. However, polynomial time algorithms that approximate the optimal solution can be designed. If such an algorithm has a proven worst-case approximation ratio, i.e., a guarantee on how far off the solution is at most, we speak of an approximation. In contrast, an algorithm that is based on a plausible idea but has no proven approximation ratio would be considered a heuristic.

Definition 2.3: Approximation Ratio. *The approximation ratio of a mechanism $\mathcal{M}(f, p)$ is defined as*

$$min_{b \in B} \frac{\mathcal{A}(b)}{OPT^*(b)}$$

where $\mathcal{A}(b)$ denotes the value of an allocation computed by an approximation algorithm $A : B \rightarrow X$ and $OPT^*(b)$ denotes an optimal allocation for a bid vector $b \in B$.

Diligently designed pricing functions can achieve further important mechanism design desiderata by providing additional (monetary) incentives for bidders. Incentives which lead to truthful behavior are among the most important ones; the resulting mechanisms are referred to as *strategyproof* or incentive-compatible. These mechanisms prevent manipulation of the outcome and relieve bidders of the burden to develop complex bidding strategies since truthful bidding is an optimal strategy. In other words, regardless of the behavior of competing bidders, truth-telling always leads to weakly higher utility than any other strategy. For a bid vector b, let b_{-i} denote the bid vector where the ith entry is removed, and by the tuple (v_i, b_{-i}), denote the bid vector where the ith entry is replaced by the true valuation of bidder i. Also, in a slight abuse of notation, let $u_i(b)$ give the utility of bidder i given bid vector b.

Definition 2.4: Strategyproofness. *A mechanism $\mathcal{M}(f, p)$ is strategyproof if for all bidders $i \in \mathcal{I}$ truth-telling leads to a weakly higher payoff for all bid vectors $b \in B$:*

$$u_i(v_i, b_{-i}) \geq u_i(b)$$

If possible, it is preferable to realize stronger notions of strategyproofness. Group-strategyproofness extends the concept such that even groups of bidders cannot benefit by colluding and deviating in a coordinated manner. By b_C, we denote the vector of bids where a coalition $C \subseteq \mathcal{I}$ deviates from truthful behavior, and by $b_{-C} = b_C \setminus \{b_i\}$, the vector without the bid of bidder $i \in C$.

Definition 2.5: Group-Strategyproofness. *A mechanism $\mathcal{M}(f,p)$ is group–strategyproof if for all coalitions of bidders $C \subseteq \mathcal{I}$ and all bid vectors $b \in B$, whenever one bidder $j \in C$ has a strictly higher utility by deviating, there is at least on other bidder $i \in C, i \neq j$ for whom truth-telling leads to a strictly higher payoff:*

$$\exists j \in C : u_j(v_j, b_{-C}) < u_i(b) \Rightarrow \exists i \in C : u_i(v_i, b_{-C}) > u_i(b)$$

Unfortunately, group-strategyproofness is difficult to achieve in many contexts. Weak group-strategyproofness, however, might still be attainable. This closely related concept is satisfied as long as deviating behavior does not lead to strictly higher utilities for every colluding bidder. In practice, this weaker notion is often sufficient due to the fact that collusion involves both trust and work - resources bidders not strictly increasing their utilities might not be willing to commit.

Definition 2.6: Weak Group-Strategyproofness. *A mechanism $\mathcal{M}(f,p)$ is weakly group-strategyproof if for all coalitions of bidders $C \subseteq \mathcal{I}$ and all bid vectors $b \in B$, truth-telling leads to a weakly higher payoff for at least one bidder $i \in C$:*

$$u_i(v_i, b_{-C}) \geq u_i(b)$$

Besides offering additional monetary benefits to achieve strategyproofness, not overcharging bidders is an equally important consideration in pricing mechanisms. Ex-ante, accurately assessing the landscape of competition can pose a challenging task for bidders. In many cases, only a fraction of the stated bids gets accepted, leaving the majority of bids unallocated. We refer to the bidders with allocated (unallocated) bids as winning (losing) bidders. Imposing a participation fee on bidders would lead to negative utilities for losing bidders and thus discourage participation. Since this would ultimately lead to less competition, which is undesirable in most contexts, participation should be free of charge. This claim for free participation can be generalized to individual rationality where prices may not exceed the respective valuations.

Definition 2.7: Individual Rationality. *A mechanism $\mathcal{M}(f,p)$ is individually rational if for all bidders $i \in \mathcal{I}$, all bid vectors $b \in B$, and all valuations v_i, the reported value of outcome $x \in X$ to bidder i is greater than or equal to the payment $p_i((b,x))$*

$$v_i(x) \geq p_i((b,x))$$

Finally, the payments (both made by buyers and collected by sellers) in a mechanism should sum up to 0. If the mechanism incurs a loss, payments need to be subsidized from an altruistic entity. On the other hand, there might be a lack of acceptance if the mechanism diverts part of the payments to an outside party. Instead, payments made by the buyers should be distributed in full among the seller(s). Note that for a single seller s_0, this usually implies $p_{s_0} = \sum_{i \in \mathcal{I}} p_i$.

Definition 2.8: Budget-Balance. *A mechanism $\mathcal{M}(f, p)$ is budget-balanced if the sum of its payments equals 0.*

To summarize, strong notions of strategyproofness in economic applications are usually realized by additional payments to the bidders. Carefully designed pricing schemes are required to compute payments that are sufficiently high without overpaying bidders or imposing fees on participation while achieving budget-balance. Chapter 3 features mechanisms which are strategyproof or even weakly group-strategyproof, including the VCG mechanism, which is the best-known example of a strategyproof mechanism.

The VCG Mechanism

The Vickrey-Clarke-Groves (VCG) mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973) is unique in the sense that it is the only strategyproof mechanism that maximizes social welfare (Green and Laffont, 1977). In a first step, the VCG mechanism determines an optimal (welfare-maximizing) allocation. Given the assumptions in Section 2.1 and using binary allocation variables $x_i(S)$ to indicate the allocation of bundle S to bidder i when $x_i(S) = 1$, this can be formulated in the *Winner Determination Problem* (WDP). The WDP is an integer linear program maximizing the social welfare subject to typical economic constraints.

$$WDP(\mathcal{I}) = \max \sum_{S \subseteq \mathcal{K}} \sum_{i \in \mathcal{I}} x_i(S) v_i(S) \tag{WDP}$$

$$\text{s.t.} \sum_{S:k \in S} \sum_{i \in \mathcal{I}} x_i(S) \leq 1, \forall k \in \mathcal{K} \tag{2.1a}$$

$$\sum_{S \in \mathcal{K}} x_i(S) \leq 1, \forall i \subseteq \mathcal{I} \tag{2.1b}$$

$$x_i(S) \in \{0, 1\}, \forall S \subseteq K, \forall i \in \mathcal{I} \tag{2.1c}$$

This standard formulation depicts an auction with a single seller and different items with a supply of 1 each. The objective function (WDP) maximizes the sum of valuations for assigned bundles. This is subject to the constraint of supply and demand (2.1a) to account for limited supply while each bidder can win at most one bid (using a XOR language, 2.1b). As mentioned before, $x_i(S)$ are binary variables (2.1c). The WDP is a pure allocation problem which does not consider prices and can serve as a basis to model a variety of related scenarios.

In the example depicted in Table 2.1, bidders 1 and 2 are interested in obtaining a single item while bidder 3 wishes to acquire both. To maximize social welfare, items A and B are allocated to bidders 1 and 2.

	Item A	Item B	Items A&B
Bidder 1	2	0	0
Bidder 2	0	2	0
Bidder 3	0	0	3

Table 2.1: Example: VCG auction

The payment that winners in a VCG auction are obliged to takes into account the stated valuation (bid) and a discount based on the respective marginal contribution to social welfare. Formally, the VCG payment can be defined as follows

$$p_i^{VCG} = b_i(S) - (WDP(\mathcal{I}) - WDP(\mathcal{I} \setminus \{i\})) \tag{2.2}$$

where $WDP(\mathcal{I})$ is the objective function value of the WDP. Note that VCG payments can only be computed if the removal of any single bidder still admits a feasible allocation and might not be budget-balanced otherwise (no single-agent effect). In the example, $WDP(\mathcal{I}) = 4(= 2 + 2)$ and $WDP(\mathcal{I} \setminus i) = 3, i \in \{1, 2\}$. This results in a discount of 1 for both allocated bidders, generating a total revenue of 2.

While the VCG mechanism offers many desirable properties such as efficiency and strategyproofness, it is no universal remedy. Firstly, it is limited to the quasi-linear setting. This implicitly assumes that market participants can pay up to their valuations, which is inappropriate in markets like the secondary market for airport time slots we consider in Chapter 4. Further, it is evident that the computation of VCG payments requires repeated solving of the allocation problem (once for the actual allocation and one additional time for each winning bidder). Depending on the problem, not even solving the original allocation to optimality might be tractable. Most commonly, problems solvable in polynomial time are considered tractable, while NP-hard problems are not. If an optimization problem can be solved in polynomial time is often determined by whether it is convex or non-convex. An optimization problem is convex if its objective function and its constraints are convex. Further, most convex problems admit efficient solution algorithms (Boyd et al., 2004). However, the WDP problem is non-convex due to the set of binary variables $x_i(S)$ (Constraints 2.1c) being inherently non-convex. In fact, even very restricted settings of the WDP are known to be NP-hard (Lehmann et al., 2006). Simply using an approximate allocation instead is not feasible, since in combination with the VCG pricing rule this does not lead to a strategyproof mechanism (Lehmann et al., 2002). As many allocation problems are NP-hard, the dependency on optimal allocations limits the applicability of the VCG mechanism in practice substantially (Rothkopf, 2007; Ausubel et al., 2006). We consider one such problem stemming from the telecommunications industry in Chapter 3. A similar problem arises when decisions regarding the allocation of bidders need to be made on incomplete information. This is the case whenever bids arrive in an online fashion; we consider an online market for road capacity in Chapter 5.

Even if the allocation problem can be solved to optimality, coalitions can manipulate the outcome. In the example, consider the bid on $\{A, B\}$ by bidder 3. This bid is pivotal for discounts of the other bidders. If the bid was lower or not stated, the remaining bidders would receive their respective item for free, contradicting group-strategyproofness. Finally, the winning bidders pay a total of 2 in our example, but bidder 3 would have been willing to pay an amount of 3. Thus, bidder 3 and the auctioneer have incentives to ignore the VCG outcome and trade among themselves. Only appropriately chosen (*core*) prices can prevent such scenarios.

The Core

Even after an allocation and prices have been computed, market participants may try to deviate. The core provides a notion of stability that prevents coalitional deviations after the market closes. Such deviations are provoked by outcomes that a subset of participants can improve upon. The example introduced in the previous section (Table 2.1) can serve as an illustration: Items are assigned to bidders 1 and 2 for a total cost of 2, while bidder 3 would have been willing to pay a total of 3. This scenario is depicted in Figure 2.1.

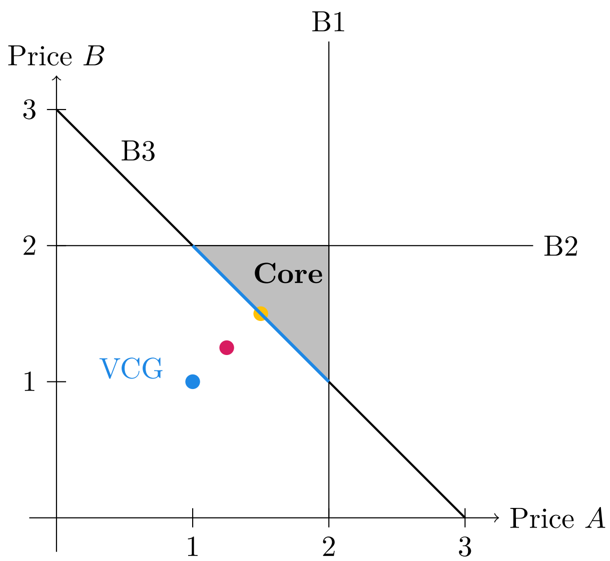


Figure 2.1: Example: the core (gray area), VCG payments (blue point), coalitional payments (red point), VCG-nearest core payments (yellow point), and bidder-optimal payments (blue line)

Such a scenario is perceived as unfair by unallocated buyers willing to pay the demanded prices (or more). By proposing a payment of 2.5 for both items, bidder 3 and the auctioneer can increase their respective utilities by at least 0.5 each (Figure 2.1, red point). We use s_0 to denote the (single) auctioneer and $\mathcal{N} = \mathcal{I} \cup \{s_0\}$ to refer to the set of bidders and the auctioneer. Also, we refer to subsets of participants as coalitions $C \subseteq \mathcal{N}$. Note that, in deviation to the definition used for (weak) group-strategyproofness, coalition C may include the single seller s_0. Forming a coalition can be viewed as a coalitional game with transferable utility (Ausubel et al., 2006). In this line, we can formally define the corresponding coalitional welfare $w(C)$ as the sum of the transferable utilities.

$$\forall C \subseteq \mathcal{N}, w(C) = \begin{cases} \max_{x \in X_C} \{\sum_{i \in C} v_j(X)\} & \text{if } s_0 \in C \\ 0 & \text{else} \end{cases} \tag{2.3}$$

where, by X_C, we denote the set of allocations feasible for coalition C. Only coalitions C including the auctioneer s_0 can actually trade; hence, the welfare is 0 if $s_0 \notin C$. If beneficial deviation for a coalition of participants is possible, we consider the proposed allocation-price combination *unstable* and refer to the coalition as *blocking*. In practice, stability of outcomes is essential since unstable outcomes would not endure but fall victim to lucrative collusion. The *core* provides a formal concept of stability.

Definition 2.9: The Core. *A core payoff vector* Π *is defined as:*

$$Core(\mathcal{N}, w) = \{\Pi \geq 0 \mid w(\mathcal{N}) = \sum_{i \in \mathcal{N}} u_i, w(C) \leq \sum_{i \in C} u_i, \forall C \subset \mathcal{N}\}$$

Under the standard assumption of transferable utilities, the outcome of a mechanism is stable if a coalition's welfare is weakly lower than the sum of transferred utilities in the proposed outcome. This definition depicts the *strong* core, while the *weak* core only requires the absence of coalitions where all members are strictly better off. When either definition is satisfied, the outcome is considered core-stable. Reconsidering our example, we can observe that assigning both items to bidder 3 would in turn be unstable against the originally proposed distribution of items to bidders 1 and 2. It is then evident that the suggested allocation allows for stable prices, but VCG pricing simply leads to prices which are too low.

Setting prices to the value of the corresponding winning bid (pay-as-bid or first-price) would give a stable outcome. In fact, using a first-price rule on a welfare maximizing outcome always leads to a stable outcome, and hence the core is never empty for auctions (Day and Milgrom, 2008). However, bidding strategies in first-price auctions are not straightforward as concealing bids can increase a bidder's margin but at the same time lowers the probability of winning. Strategies are known for a few simple settings such as single-item auctions (Krishna, 2009) but are hard to determine for other scenarios. In general, first-price auctions are not strategyproof.

It has been shown that, if the VCG outcome is not in the core, no truthful, core-selecting auction exists (Goeree and Lien, 2016). Still, core-stable outcomes can be computed. However, since the core usually admits several outcomes, it is not obvious how to rank outcomes, let alone choose one. Bidder-optimal core prices minimize the sum of payments transferred to the auctioneer and have been proposed to limit the participants' ability to manipulate the outcome (Day and Raghavan, 2007). The blue line in Figure 2.1 indicates the several different combinations of prices minimizing the seller revenue among the core outcomes, illustrating that this still does not lead to a unique outcome in general. Parkes et al. (2001) argue that the difference between the computed payment and the VCG payment poses a "residual incentive to misreport" and should thus be minimized (Figure 2.1, yellow point). Day and Cramton (2012) suggest an approach minimizing the sum of squared deviations from a reference point (e.g., the VCG point, Figure 2.1). These and closely related results have been applied to spectrum auctions (e.g., in Cramton (2013); Goetzendorff et al. (2015)), auctions of natural resources (Cramton, 2010), and the allocation of virtual machines in cloud computing (Fu et al., 2014). Open questions remain in regard to combinatorial exchanges for which Bikhchandani and Ostroy (2002) have shown that, contrarily to auctions, the core can be empty. We present results on computing welfare-maximizing core-stable outcomes in combinatorial exchanges in Chapter 4 and introduce the necessary prerequisites in 2.3.1.

2.3 Beyond Standard Assumptions

In some applications, not all the assumptions made in traditional combinatorial auction literature translate to practice. In advertisement markets (Bichler and Merting, 2018) and markets where buyers are financially constrained (Chapter 4), utilities are not quasi-linear. Furthermore, online markets demand prices to be computed dynamically on incomplete information (Chapter 5). In both markets, the VCG mechanism is not applicable due to the non-quasi-linear setting or incomplete information inhibiting the computation of welfare-maximizing allocations. This section gives a short introduction to these markets beyond standard assumptions, the respective literature, and corresponding solution strategies.

2.3.1 Non-quasi-linear Markets

It is not always realistic to assume that bidders can pay up to their valuation, as in many industries bidders are financially constrained. The market we consider in the second project (Chapter 4) can serve as an example, as it considers the very cost intensive airline industry. One could deal with budget constraints by having bidders not state their valuation but the minimum out of their valuation and budget (a *capped bid*) instead. However, this results in incomplete information constituting the risk of inefficient or unstable outcomes as we show in Chapter 4. Alternatively, the mechanism might allow bidders to state their valuations and budgets separately as done in practice for auctions for online (Balseiro et al., 2015) and TV ads (Nisan et al., 2009). It is important to note that in the presence of budget constraints, the assumption of quasi-linear, i.e., fully transferable, valuations does not hold. Thus, the VCG mechanism is inapplicable, and no strategyproofness mechanism can be designed (Dobzinski et al., 2012). At the same time, the ability of individuals to influence the outcome dwindles in large markets with many participants (Roberts and Postlewaite, 1976): Bidders become price-takers and strategyproofness becomes a second-order objective. In this light, another design desideratum becomes crucial: core stability.

While core-stable outcomes always exist for auctions (Milgrom, 2004), the same is not true for exchanges. In an exchange, multiple buyers $\mathcal{I}$ and multiple sellers $\mathcal{J}$ trade on the same market. Stable allocations are harder to achieve here since no single, in regard to stability pivotal, auctioneer exists. In fact, for combinatorial exchanges, the core can be empty even in the absence of budget constraints. An example is given in Table 2.2 where two bidders compete over items A and B offered by two sellers. Whenever a single item is allocated to bidder 1, bidder 2 can form a coalition with the sellers by offering a price > 2 for the item assigned to bidder 1. However, allocating the package AB to bidder 2 is not stable either, as there will always be (at least) one item with a price < 2, i.e., an opportunity for bidder 1 to outbid bidder 2 on this item. Due to the cyclic relationship of the blocking coalitions, no stable outcome can be computed and the core is empty.

	Item A (Seller 1)	Item B (Seller 2)	Items A&B (Sellers 1&2)
Bidder 1	2	2	0
Bidder 2	0	0	3

Table 2.2: Example: Combinatorial exchange with empty core

Evidently, assigning trivial budgets (up to the bidders' respective valuations) gives an example where the core is empty for a combinatorial exchange with budget constraints. Still, in general, one cannot simply ignore budgets and draw conclusions for the scenario with budget constraints. Bichler and Waldherr (2019) have shown that the core can be empty for one scenario but non-empty for the other. For illustration, we restate the proof of Bichler and Waldherr (2019) below:

Assigning the bundle AB to bidder 1 is welfare-maximizing and stable when ignoring budgets. However, the core is empty when taking budgets into consideration. The allocation problem in combinatorial exchanges without budgets is already NP-hard, but it has been shown that computing welfare-maximizing core-stable solutions for combinatorial exchanges with budgets constraints is even $\sum_2^{\mathrm{P}}$-hard (Bichler and Waldherr, 2019).

	Item A (Seller 1)	Item B (Seller 2)	Items A&B (Sellers 1&2)	Budgets
Bidder 1	0	0	10	3
Bidder 2	4	4	0	2

Table 2.3: Example: Combinatorial exchange with non-empty core that is empty when considering budgets (taken from: Bichler and Waldherr (2019))

This complexity class is one step above NP in the polynomial hierarchy, i.e., if there was an oracle that could solve problems in NP in constant time, the problem would still be NP-hard. Motivated by an application in airport time slot markets, our second project (Chapter 4) is concerned with the Σ_2^{P}-hard problem of computing welfare-maximizing core-stable outcomes in combinatorial exchanges with budget-constrained bidders. Despite its inherent complexity, we can solve instances of relevant size using an approach based on bilevel programming.

Bilevel Programming

Only few solution strategies exist for problems in the complexity class of Σ_2^{P}. Mixed-integer bilevel linear programs pose one possible way to formulate problems in this class. A bilevel linear program is an optimization problem constrained by another optimization problem. The general form of a mixed-integer bilevel linear problem is depicted below.

$$\min\nolimits_{x \in X} \; F(x,y) \tag{2.4a}$$
$$\text{s.t. } G(x,y) \leq 0 \tag{2.4b}$$
$$\min\nolimits_{y \in Y} \; f(x,y) \tag{2.4c}$$
$$\text{s.t. } g(x,y) \leq 0 \tag{2.4d}$$

where $F, f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^1, G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q, g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ are continuous, twice-differentiable linear functions and X, Y are sets of decision variables, which can be continuous or integer. One can think of bilevel programs as a game of leader and follower: The leader solves the upper level (UL) problem with objective function 2.4a subject to constraints 2.4b and variables x. The UL variables are then fixed in the lower level problem (LL) (the follower "observes" the leader's decision). Subsequently, the follower optimizes objective 2.4c given x and constraints 2.4d. The decision variables y of the LL are then passed back to the leader. Finally, the follower's decision is reflected in the objective function or the constraints of the UL, worsening the objective function value or rendering the solution infeasible.

Depending on whether variables are continuous, integer, or both in the respective levels, different solution strategies can be employed. If the LL is purely continuous, the problem can be re-formulated into a single level problem. This is most often done by replacing the LL by the respective Karush-Kuhn-Tucker (KKT) conditions (optimality conditions) in the UL (Karush, 1939; Kuhn and Tucker, 2014). If the linking variables, i.e., variables that appear in both the UL and the LL, are integer, the convex hull of the feasible set is closed, and at least one solution exists. Fischetti et al. (2017) and Tahernejad et al. (2017) developed algorithms based on this observation. The mixed-integer case, where both levels have integer as well as continuous variables, has received less algorithmic proposals. Depending on the context, meta-heuristics such as differential evolution (Koh, 2013) or tabu search (Rajesh et al., 2003) could be applied, but the non-deterministic nature of these approaches rules them out for most applications in markets. They also experience surprisingly long runtimes if the LL is hard to solve since all randomly generated points need to be evaluated, which requires solving the LL repeatedly. A single-level reformulation for special cases where the LL is independent of the UL was suggested by Dempe and Franke (2014). For the same class of problems, Jeyakumar et al. (2016) suggested an algorithm based on semidefinite relaxations. There is, however, only little literature on the general case.

The lack of algorithms for general mixed-integer bilevel programming is arguably due to the fact that many techniques known to be helpful in the context of solving single-level mixed-integer programs do not translate to the bilevel case (Dempe, 2002). Relaxing integer variables in the LL does not lead to a lower, but rather an upper bound of the bilevel (minimization) problem. Also, constructing a convergent lower bounding problem is notoriously difficult (Mitsos and Barton, 2006). Finally, partitioning (branching on) the space of the LL can lead to invalid bounds and the selection of infeasible solutions. It has been argued that algorithms should always consider the whole LL domain (Mitsos et al., 2008), but literature on branching the LL is sparse.

When solving general mixed-integer bilevel programs, one can adopt one of two perspectives if multiple optimal solutions for the LL exist. In the optimistic perspective, which the majority of research follows, one assumes that out of the LL-optimal solutions the solution leading to the best UL objective function value is returned. Contrarily, one believes the worst LL-optimal solution (in terms of UL objective function value) is chosen in the pessimistic perspective (Bard, 2013). Based on the optimistic formulation, Kleniati and Adjiman (2014) developed a branch-and-sandwich algorithm for general, mixed-integer (non-linear) bilevel programs. Their algorithm allows branching in both (upper and lower) domains using an elaborate system of lists to manage nodes. Improvements have been made over a series of subsequent papers (Kleniati and Adjiman, 2015; Paulavičius and Adjiman, 2020) with the computational studies available in Paulavičius et al. (2016) and Paulavičius et al. (2020).[3] The approach heavily relies on overestimations and reformulations and is fairly challenging to implement.

A simpler method was proposed by Zeng and An (2014). Their algorithm further constraints the UL problem in every iteration by adding KKT conditions for the LL problem. However, KKT conditions are only applicable if the problem they are based on is convex, and thus values of integer variables need to be fixed in the LL. In the worst-case, this deteriorates to an enumeration approach but has shown promising results in a computational study (Zeng and An, 2014). It also stands out due to its simplicity and mild assumptions, making it a very applicable framework we use in Chapter 4.

2.3.2 Online Markets

In the context of combinatorial auctions, it is standard to assume that all bids are available when the outcome is computed, but this is not necessarily the case. In fact, our third project (Chapter 5) is concerned with an application in congestion pricing where drivers arrive one by one. We refer to problems where at the decision point only incomplete information is available as *online problems*. Dynamic pricing constitutes a prominent example, as many economic applications can be related to this (e-commerce, booking of hotels or flights, and car rental) (Gallego and Van Ryzin, 1994). In this problem, a seller aims to maximizes revenue over a finite time horizon given a finite

[3] Just recently, an implementation of the branch-and-sandwich algorithm was made available in the Minotaur toolkit: `https://wiki.mcs.anl.gov/minotaur/index.php/Running_Binaries` (Paulavičius and Adjiman, 2019). Our work in Chapter 4 precedes this implementation.

inventory of items. The corresponding offline problem can be modeled as slight variation of the WDP:

$$\max \sum_{S\subseteq\mathcal{K}} \sum_{i\in\mathcal{I}} x_i(S) v_i(S) \tag{2.5a}$$

$$\text{s.t.} \sum_{S\subseteq\mathcal{K}} \sum_{i\in\mathcal{I}} a_i x_i(S) \leq c(k), \forall k \in \mathcal{K} \tag{2.5b}$$

$$\sum_{S\in\mathcal{K}} x_i(S) \leq 1, \forall i \subseteq \mathcal{I} \tag{2.5c}$$

$$x_i(S) \in \{0,1\}, \forall S \subseteq K, \forall i \in \mathcal{I} \tag{2.5d}$$

where $c(k) \in \mathbb{R}_0^+$ denotes the inventory available of item $k \in \mathcal{K}$ and vector $a_i \in \{0,1\}^m$ denotes the items requested in bid $x_i(S)$. In the online problem, inventories $c(k)$ are known at the beginning for all m items $k \in \mathcal{K}$, but the n requests (bids), i.e., coefficients $v_i(S)$ and columns a_i of the constraint matrix, arrive one by one. The corresponding decision on accepting or rejecting this request is due immediately: the seller needs to choose $x_i(S)$ such that the objective function value is maximized at the end of the time period while adhering to inventory constraints. Usually, the decision is based on availability of the requested items and/or a comparison of the coefficient $v_i(S)$ against some threshold value (e.g., a sum of item prices) which is learned as requests arrive over time (Agrawal et al., 2014).

To assess the performance of online algorithms, two measures have developed mostly independently in fields closely related to online linear programming (Andrew et al., 2013). Metrical task systems (MTS) decide on a sequence of actions in an online fashion, incurring costs for the actions themselves and switching costs between them (defined on a metric), aiming to minimize the total costs. The predominant measurement in MTS is the competitive ratio, i.e., the ratio of the solution computed by the online algorithm and an optimal *dynamic* offline solution (e.g., Coté et al. (2008)). An optimal dynamic offline solution is computed on the complete set of requests (including their ordering) and can thus be determined optimally.

Definition 2.10: Competitive Ratio. *The competitive ratio of an online allocation algorithm* $f_{online} : B \rightarrow X$ *is defined as*

$$\frac{f_{online}(b)}{OPT_d^*} \tag{2.6}$$

where OPT_d^* *is an optimal dynamic offline solution.*

Online convex optimization (OCO), where action spaces and valuation (traditionally considered as cost) functions are convex, does not consider switching costs. Furthermore, it is usually assessed using *regret* instead of the competitive ratio (e.g., Herbster and Warmuth (1998)). Regret is defined as the difference between an optimal *static* offline solution and the solution computed by the online algorithm. Note that while an optimal static offline solution has knowledge of all requests at the point of decision (offline), the action cannot be changed in between requests (static). For a maximization problem, it can be formalized as follows:

Definition 2.11: Regret. *The regret of an online allocation algorithm* $f_{online} : B \rightarrow X$ *is defined as*

$$Regret = OPT_s^* - f_{online}(b) \tag{2.7}$$

where OPT_s^* *is an optimal static offline solution.*

When analyzing online optimization problems under the traditional assumption of worst-case input (Borodin and El-Yaniv, 2005), the performance bound is very pessimistic as only $\mathcal{O}(1/n)$ of the optimal objective function value can be attained by any online algorithm (Babaioff et al., 2008). The proof can be sketched in the following way: for a single resource with an inventory of 1 and a sequence of requests $K^0, K^1, K^2, \ldots, K^m, 1, 1, \ldots, 1$ with $K >> n$, m can be chosen in such a way that any algorithm will choose a request other than K^m and thus achieves an object function value of at most $1/n$ of the optimum. On the other hand, assuming independent and identical distributed (i.i.d.) requests according to a known distribution, the optimal strategy can be determined in polynomial time (Babaioff et al., 2008).

In the search for a middle ground between worst-case analysis and known distribution, two important models have emerged: stochastic input and random permutation. In the *stochastic input model*, one assumes that columns and corresponding coefficients are i.i.d. drawn from an unknown distribution. In contrast, the *random permutation model* assumes columns and coefficients arriving in random permuted order with uniform distribution over all permutations. Note that values can be chosen to be adversarial in the random permutation model. In particular, the random permutation model is weaker in its assumptions (Mehta, 2013) and subsumes the stochastic input model (Babaioff et al., 2008). The random permutation model is thus considered more suitable to depict the possibility of heterogeneous, non-stationary, or unfavorable demand (Li et al., 2020; Agrawal et al., 2014).

There are algorithms that (in expectation and under the assumptions of one of the models) achieve a constant-factor competitive ratio (Lin et al., 2012) or sub-linear regret ("no regret") (Zinkevich, 2003) for certain problems, but Andrew et al. (2013) have shown that no algorithm can achieve both. Recently, Agrawal et al. (2014) have established that the necessary right hand-side condition for a $(1-\epsilon)$-competitive algorithm is $c \in \Omega(\frac{\log m}{\epsilon^2})$. The sufficient part has been completed in subsequent work by Kesselheim et al. (2014) and Gupta and Molinaro (2014) who developed $(1-\epsilon)$-competitive algorithms based on the necessary condition. In a similar fashion, Li and Ye (2019) have investigated regret under the condition of the right-hand side growing linearly in the number of requests and derived a regret bound of $\mathcal{O}(\log(n)\log(\log(n)))$.

In online integer linear programming, the action space is not convex and possibly very limited if decision variables are binary. In dynamic pricing, the action space usually only contains the binary decision to either accept or reject the request. The standard definition of regret stated above becomes less meaningful then, as an optimal static offline solution might reject every request (to not violate capacity constraints). To overcome this limitation, *expected dynamic regret* is used instead (e.g., Li et al. (2020)). Departing from the definition of regret used in OCO, expected regret is defined on the optimal *dynamic* offline solution and, eponymously, on (worst-case) expectation:

Definition 2.12: Expected Regret. *In online linear programming, expected regret of an online algorithm* $f_{online} : B \rightarrow X$ *is defined as*

$$max_{\mathcal{P}\in\Xi} \left(\mathbb{E}_{\mathcal{P}}\left[OPT_d^* - f_{online}\right]\right) \tag{2.8}$$

where $\mathcal{P}$ *is an input distribution,* Ξ *is the set of input distributions according to the assumptions of the model,* $\mathbb{E}_{\mathcal{P}}$ *is the expectation defined on* $\mathcal{P}$*, and* OPT_d^* *is an optimal dynamic solution.*

Since, in the context of online integer linear programming, solutions are not feasible in general, *constraint violation* has been established as an additional metric.

Definition 2.13: Constraint violation. *Constraint violation of an online linear program is defined as the 2-norm on the constraint matrix A, the decision vector x, and the right-hand side c:*

$$||(Ax - c)^+||_2 \tag{2.9}$$

where $f(x)^+$ denotes the positive part of a function $f(x)$ and $||x|||_2$ is the Euclidean norm, i.e., $||x|||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ for a vector x of dimension n.

Finally, the capability to handle large inputs is often considered crucial in online algorithms (Agrawal et al., 2014; Li et al., 2020). Li et al. (2020) developed an algorithm which can handle large inputs due to fast computation of updates, achieves expected (dynamic) regret and constraint violation of $\mathcal{O}(m\sqrt{n})$, and is minimal in its assumptions. We apply this algorithm to a problem in congestion management in Chapter 5.

2.4 Real-World Applications

In this section, we introduce the real-world applications on which the following chapters are based on. We give a brief introduction to the respective application, provide a short literature review, and point out complications of practical relevance.

2.4.1 Steiner Trees in Telecommunications

The telecommunications industry generates revenue by operating and providing access to networks such as the mobile network, landlines, or the Internet. The perceived service quality heavily depends on network reliability and speed, making efficiently designed networks a necessity. A typically network design problem in telecommunications is to establish connectivity between essential nodes at minimal cost, while the inclusion of further nodes is optional. Motivated by a real-world application (subject to NDAs), we consider a reverse auction where a single auctioneer buys connections between nodes from suppliers with quasi-linear utilities. Thus, strategyproof mechanisms are essential to prevent manipulation in this setting.

The underlying allocation problem corresponds to the Steiner minimum tree (SMT) problem, considered "*the* combinatorial optimization problem in telecommunications" (Voß, 2006). Outside of the game-theoretical setting, the SMT problem has received a lot of attention as an optimization problem with various industry applications (Cheng and Du, 2013). In general, the SMT problem is NP-hard to approximate within 96/95 of the optimal solution (Chlebík and Chlebíková, 2008) and is in fact one of Karp's 21 NP-complete problems (Karp, 1972). Researchers suggested exact approaches based on dynamic programming (Dreyfus and Wagner, 1971; Erickson et al., 1987; Fuchs et al., 2007; Mölle et al., 2006) and branch-and-cut (Lucena and Beasley, 1998; Chopra et al., 1992; Koch and Martin, 1998). As the associated runtimes are exponential in the number of terminals, these algorithms are impractical when inputs grow large. Incidentally, this rules out application of the VCG mechanism for practical applications.

Numerous approximations for the SMT problem are available (Berman and Ramaiyer (1994); Zelikovsky (1996); Prömel and Steger (2000); Karpinski and Zelikovsky (1997), and (Hougardy and Prömel, 1999)) but cannot guarantee strategyproofness in combination with VCG payments, and despite the need for strategyproof mechanisms in network design markets, there is only little prior work (Gualà and Proietti, 2005). Gualà and Proietti assume single-minded bidders, i.e., suppliers that offer exactly one edge. We adopt this assumption, which is realistic in the telecommunications setting, and, building on results by Myerson (1981), provide further strategyproof approximation mechanisms for the SMT problem. Additionally, we consider deferred-acceptance auctions (DAA), a general class of weakly group-strategyproof mechanisms. There is no known approximation bound for DAAs on the SMT problem, and the stronger notion comes at the price of additional payments to the suppliers. However, the results of a computational study show that simple DAA variants find better allocations than dedicated SMT approximations, and the premium associated with weak group-strategyproofness is relatively low (Chapter 3).

2.4.2 Airport Time Slots

Prior to (and most likely after) the COVID-19 pandemic, demand at virtually all major European airports exceeded available capacities. Different mechanisms battling airport congestion have been installed, with the IATA slot allocation system being the most widely used (Ulrich, 2008). Under IATA regulations, time slots (rights to take-off or land at a certain airport at a certain time) are assigned biannually to airlines (primary markets). Grandfathering rights enable incumbents to keep the market share of entrants low by blocking them from acquiring a meaningful number of slots. Thus, primary markets are currently dominated by grandfathering rights, and although secondary markets (trades among airlines) with monetary transfers are permitted in the EU, the acceptance has been low. While there is consensus that current practices are inefficient, suggestions for improvement range from enhancing allocations in primary markets (Zografos et al., 2012; Corolli et al., 2014; Pellegrini et al., 2017) to abolishing grandfathering rights altogether (Ewers, 2001). Since the IATA clearly stated "[they] would oppose any consideration of market-based primary slot allocation mechanisms" (Organization, 2016), focusing on secondary markets seems to be the more promising way going forward.

Secondary markets have been identified as a beneficial addition to primary allocations (Ltd., 2001; MacDonald, 2006) to compensate for slot complementarity problems (Madas and Zografos, 2013), but the current implementation is hindered by the need of bilateral negotiations. Recent proposals have sought to overcome this by implementing combinatorial exchanges (Pellegrini et al., 2012). In our second project, we propose a combinatorial exchange with budget constraints, which have not been previously considered despite the cost intensity of the airline industry (Chapter 4). Under budget constraints, utility functions are no longer quasi-linear, as only part of the utility can be transferred by payments. As the VCG mechanism is designed for quasi-linear markets, it is not applicable here. In fact, no mechanism can achieve strategyproofness in this setting (Dobzinski et al., 2012). We instead aim for core stable outcomes to ensure resilience of the outcome after the market closes. Computing welfare-maximizing core-stable outcomes in combinatorial exchanges is Σ_2^{P}-hard (Bichler and Waldherr, 2019), which is usually considered intractable. We use a bilevel mixed-integer linear program to model the problem and apply an algorithm by Zeng and An (2014) to solve it. We replace the general purpose KKT constraints used in the algorithm by a set of constraints representing the sum of transferable utility (similar to Definition 2.9). This yields large performance improvements, which permit solving instances of relevant size despite the inherent complexity of the problem. Further results underline the importance of budget constraints by providing evidence that solutions are not welfare-maximizing or even unstable when budgets are ignored.

2.4.3 Congestion Pricing for Road Networks

Urban areas all over the world suffer from congested roads, especially on highways and in city centers. On a societal level, this leads to detrimental effects on the environment and the economy. In their 2019 report, the Texas Transportation Institute reports a 15% increase in yearly delay per car commuter and even a 19% increase in congestion cost between 2012 and 2017 in the USA.[4] Congestion is a concise example of the *tragedy of the commons*. In this allegory, more and more people make use of a free resource (a road) until a state of overuse (congestion) is reached. Consequently, the usefulness (travel speed) of the resource either decreases significantly or is even nullified. In other words, free-riding behavior by individuals does not take into account the caused externalities regarding congestion, making society as a whole worse off. *Marginal social pricing* is based on the idea that the externalities a participant causes should be paid for by that participant (Pigou, 1920). While not specific to it, this laid the foundation for congestion pricing, with Nobel laureate William Vickrey (1952) being the first to argue in its favor.

Modern congestion pricing schemes differ in two main dimensions: how payments are registered and how prices are adapted. Along the first dimension, schemes defined on facilities, cordons, areas, and distance can be distinguished. In a facility-based scheme, simple entities such as single roads, tunnels, or bridges are priced (e.g., the Channel Tunnel[5]). Cordon-based schemes require payment whenever a driver crosses a cordon (in- and/or outbound). For example, several Norwegian cities implemented cordon-based systems and Milan introduced one in 2008 (mainly to reduce emissions) (Rotaris et al., 2010). Similarly, pricing mechanisms that are based on an area (defined by either natural or artificial borders) charge customers for entering the area and starting trips in it. The London congestion charge in the central city area provides a practical example.[6] Finally, distance-based schemes charge travelers based on the distance traveled on priced facilities. Many European countries introduced such schemes for heavy vehicles with the goal of recovering cost related to road erosion (de Palma and Lindsey, 2011). From a theoretical point of view, road-level prices should be preferred since they correspond to the ideal of Pigouvian prices (de Palma and Lindsey, 2011).

Congestion pricing schemes can further be distinguished by responsiveness of prices. Static schemes charge pre-determined prices at all times and do not adapt prices based on time of day and/or congestion levels. In this class of pricing schemes, prices can change regularly but always follow a pre-determined pattern that is non-responsive to congestion. Examples include the Øresund Bridge[7] (no differentiation) and the London

[4] `https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf`, accessed: 02.11.2020

[5] `https://www.eurotunnel.com/uk/`, accessed: 28.11.2020

[6] `https://www.bbc.com/news/uk-england-london-52677059`, accessed: 02.11.2020

[7] `https://www.oresundsbron.com/en/prices`, accessed: 30.11.2020

congestion charge (time-of-day differentiation only). Contrarily, responsive (dynamic) pricing schemes adopt prices based on the current traffic situation. These schemes are more potent than non-responsive schemes, as they can reproduce static behavior while also providing options to adapt prices more freely based on demand. In particular, they more accurately represent marginal social pricing (Cheng et al., 2017). So far, only trials of dynamic pricing schemes have been implemented in practice (Cramton et al., 2017). Finally, predictive pricing schemes set prices according to predicted instead of observed demand. However, predictive pricing schemes have only been considered in very restricted setting with one road (Yin and Lou, 2009; Lou et al., 2011; Lu and Zhou, 2014), as the field is still in its infancy. Thus, dynamic pricing schemes represent the most practical way to reduce congestion in major cities.

While trial runs in dynamic road pricing have shown promising results (Cramton et al., 2017), implementing network-wide dynamic pricing schemes remains an open question (Cheng et al., 2017). To adhere to marginal social pricing, prices need to be updated in real-time. Thus, fast price updates are obligatory although information regarding future demand is sparse. Due to the online nature of the problem, the VCG mechanism cannot be applied, as no welfare-maximizing allocation can be computed based on incomplete information.

Under the assumption of quasi-linear utility, computing prices such that an (approximately) welfare-maximizing fraction of the drivers is admitted to the road is similar to pricing scarce resources in order to maximize revenue. Thus, the optimization problem can be formulated as an online linear program where drivers arrive one by one and road capacity is the scare resource (the online variant of 2.5a). In our third project, we apply an algorithm by Li et al. (2020) to solve the online linear programming formulation (Chapter 5). The algorithm achieves expected regret of $\mathcal{O}(m\sqrt{n})$ and $\mathcal{O}((m+\log n)\sqrt{n})$ in the stochastic input and random permutation model, respectively. In a computational study on the traffic simulation SimMobility (Adnan et al., 2016), we show that dynamic road prices computed by an online linear program can more effectively reduce congestion and achieve higher welfare than static prices. Importantly, a large share of car trips is admitted, i.e., the prices lead to more efficient rather than less traffic flow by discouraging drivers through high prices. The model only assumes quasi-linear utilities, known road capacities, and availability of predictions for the number of requests in a time interval and thus provides valuable insights for practical applications.

3 Strageyproof Pricing

Peer-Reviewed Journal Paper

Title: Strategyproof auction mechanisms for network procurement

Authors: M. Bichler, Z. Hao, R. Littmann, S. Waldherr

In: OR Spectrum

Abstract: Deferred-acceptance auctions can be seen as heuristic algorithms to solve $\mathcal{NP}$-hard allocation problems. Such auctions have been used in the context of the Incentive Auction by the US Federal Communications Commission in 2017, and they have remarkable incentive properties. Besides being strategyproof, they also prevent collusion among participants. Unfortunately, the worst-case approximation ratio of these algorithms is very low in general, but it was observed that they lead to near optimal solutions in experiments on the specific allocation problem of the Incentive Auction. In this work, which is inspired by the telecommunications industry, we focus on a strategic version of the minimum Steiner tree problem, where the edges are owned by bidders with private costs. We design several deferred-acceptance auctions (DAAs) and compare their performance to the Vickrey-Clarke-Groves (VCG) mechanism as well as several other approximation mechanisms. We observe that, even for medium-sized inputs, the VCG mechanism experiences impractical runtimes and that the DAAs match the approximation ratios of even the best strategy-proof mechanisms in the average case. We thus provide another example of an important practical mechanism design problem, where empirics suggest that carefully designed deferred-acceptance auctions with their superior incentive properties need not come at a cost in terms of allocative efficiency. Our experiments provide insights into the trade-off between solution quality and runtime and into the additional premium to be paid in DAAs to gain weak group-strategyproofness rather than just strategyproofness.

Contribution of thesis author: Results, implementation, presentation, project and paper management

Reference: Bichler et al. (2020a), conference version: Bichler et al. (2020b)

[8]https://creativecommons.org/licenses/by/4.0/

OR Spectrum (2020) 42:965–994
https://doi.org/10.1007/s00291-020-00597-7

REGULAR ARTICLE

Strategyproof auction mechanisms for network procurement

Martin Bichler[1] · **Zhen Hao**[1] · **Richard Littmann**[1,2] · **Stefan Waldherr**[1]

Received: 13 March 2018 / Accepted: 10 July 2020 / Published online: 25 July 2020

Abstract

Deferred-acceptance auctions can be seen as heuristic algorithms to solve $\mathcal{NP}$-hard allocation problems. Such auctions have been used in the context of the Incentive Auction by the US Federal Communications Commission in 2017, and they have remarkable incentive properties. Besides being strategyproof, they also prevent collusion among participants. Unfortunately, the worst-case approximation ratio of these algorithms is very low in general, but it was observed that they lead to near-optimal solutions in experiments on the specific allocation problem of the Incentive Auction. In this work, which is inspired by the telecommunications industry, we focus on a strategic version of the minimum Steiner tree problem, where the edges are owned by bidders with private costs. We design several deferred-acceptance auctions (DAAs) and compare their performance to the Vickrey–Clarke–Groves (VCG) mechanism as well as several other approximation mechanisms. We observe that, even for medium-sized inputs, the VCG mechanisms experiences impractical runtimes and that the DAAs match the approximation ratios of even the best strategy-proof mechanisms in the average case. We thus provide another example of an important practical mechanism design problem, where empirics suggest that carefully designed deferred-acceptance auctions with their superior incentive properties need not come at a cost in terms of allocative efficiency. Our experiments provide insights into the trade-off between solution quality and runtime and into the additional premium to be paid in DAAs to gain weak group-strategyproofness rather than just strategyproofness.

Keywords Steiner tree problem · Network procurement · Approximation mechanism · Deferred-acceptance auction

✉ Martin Bichler
martin.bichler@in.tum.de

Extended author information available on the last page of the article

1 Introduction

There is a significant literature in the design of approximation algorithms for computationally hard problems (Vazirani 2013). Algorithmic mechanism design extends this literature in an important way (Nisan and Ronen 1999). The goal of approximation mechanisms is the design of computationally efficient algorithms which take into account the incentives of participants as well. These mechanisms should run in polynomial time and satisfy strong game-theoretical equilibrium solution concepts such that bidders have incentives to reveal their valuations truthfully and the auctioneer can determine the optimal allocation or one that approximates the optimal solution. This has led to a rich literature studying approximation mechanisms for different types of $\mathcal{NP}$-hard resource allocation problems. Typically, designers of approximation mechanisms aim for dominant-strategy incentive-compatibility or strategyproofness. Such mechanisms are prior-free, and truthful bidding is a dominant strategy for individual bidders.

Network procurement is a prime application where auction mechanisms play an important role in business practice. A telecom is interested in connecting several sites or terminals via a cost-minimal set of edges connecting vertices in a network. The terminals constitute a subset of all vertices in the network, and suppliers can provide individual edges in the network at a certain cost. The minimum Steiner tree problem is a well-known model of this network procurement problem, and even with complete information about suppliers' costs, finding a cost-minimal solution is $\mathcal{NP}$-hard. The minimum Steiner tree problem on graphs is one of the most well-known $\mathcal{NP}$-complete problems (Karp 1972), and central in various types of network design problems, which have received significant attention in operations research (Xu et al. 1995; Öncan et al. 2008; Contreras and Fernández 2012).

In the procurement environment, the cost of establishing a link is the private information of its supplier. Each supplier wants to maximize her payoff, i.e. her bids minus their private costs for setting up the connection. In such an auction, the auctioneer wants to set incentives for bidders to reveal their costs truthfully. It is well known that the Vickrey–Clarke–Groves (VCG) (Vickrey 1961; Clarke 1971; Groves 1973) mechanism is the only quasi-linear mechanism which maximizes social welfare and is strategyproof (Green and Laffont 1977). Still, the resulting discounts can be manipulable by coalitions of suppliers, a property which can well be a problem in procurement. This means the VCG mechanism is not group-strategyproof. In addition, the VCG mechanism is no longer strategyproof if the allocation does not maximize social welfare, i.e. if the allocation cannot be solved exactly. Since the minimum Steiner tree problem is $\mathcal{NP}$-complete, its optimal solution, which corresponds to the maximally achievable social welfare, cannot be expected to be obtained in reasonable time.

If the allocation cannot be computed optimally, but only approximately, then the VCG mechanism loses this strong game-theoretical property (Lehmann et al. 2002). This paper analyzes several well-known approximation algorithms for the minimum Steiner tree problem with respect to their implementability in settings where the edges of the graph are strategic agents. Based on well-known theory from mechanism design, we verify that some of these approximation algorithms can be extended to strategyproof mechanisms, while others are not.

Motivated by the Incentive Auction of the US Federal Communications Commission (FCC), Milgrom and Segal (2019) and Leyton-Brown et al. (2017) recently proposed *deferred-acceptance auctions (DAAs)*, a class of greedy algorithms which are weakly group-strategyproof for bidders with single-dimensional types. This means even a coalition of bidders cannot manipulate profitably via deviations from truthful bidding, which makes them robust against collusive bidding strategies. This is a very desirable property in many applications. Also, a deferred-acceptance auction can be implemented both as a sealed-bid and as a clock auction.

An important question is whether these strong incentive properties are at the expense of solution quality, i.e. they might lead to low allocative efficiency. Dütting et al. (2017) derived worst-case approximation ratios for two important problem classes. Still, for most problems no worst-case approximation ratios have been proven. Interestingly, experimental analysis of the specific allocation problem in the US FCC Incentive Auction showed very high solution quality on average (Newman et al. 2017). In their simulations, which focused on the efficiency of the reverse auction, the reverse clock auction achieved highly efficient solutions. The specific scoring rule by the FCC played an important role in the solution quality and the payments computed. The allocation problem in the Incentive Auction is special, and it is not clear whether one could achieve high average efficiency with a DAA also for other problems.

We perform a thorough computational study in which we compare DAA variants to more sophisticated approximation mechanisms for the Steiner minimum tree problem. The results show that in general, the DAA (with an adequately chosen scoring function) results in high solution quality, but that in environments with a very sparse network and few terminals, primal-dual algorithms or Mehlhorn's algorithm is better. All approximation algorithms and heuristics were computed within only two minutes on average, while the computation times for exact solutions with a Vickrey–Clarke–Groves payment rule are extensive and took more than 18 hours on average for the larger instances. The revenue is lowest in the Vickrey–Clarke–Groves mechanism. The DAA variants led to higher payments for the buyer, which can be seen as a premium paid for group-strategyproofness, i.e. its robustness to collusion. Our empirical results illustrate the order of magnitude of these trade-offs.

In Sect. 2, we introduce related literature, before we introduce the minimum Steiner tree and relevant definitions in Sect. 3. In Sect. 4, we analyze the implementability of well-known approximation algorithms for the minimum Steiner tree problem, and a critical payment scheme, before we introduce deferred-acceptance auctions. Then, in Sect. 6 the results of numerical experiments based on the SteinLib are presented.

2 Related literature

The minimum Steiner tree problem has many important applications in a variety of fields. Examples include biology (phylogenetic trees), the design of integrated circuits, and it occurs as a special case or subproblem in many other problems in the field of network design (single-sink rent-or-buy, prize-collecting Steiner tree, single-sink buy-at-bulk). Due to its relevance, the problem received a lot of attention and different classes of algorithms emerged.

Approximation algorithms based on distance networks were proposed by Takahashi and Matsuyama (1980) and Kou et al. (1981). Mehlhorn (1988) developed a faster variant of the latter algorithm. All algorithms in this class achieve an approximation ratio of 2, which is also achievable by means of primal-dual algorithms, see e.g. Goemans and Williamson (1995). Loss-contracting approximations are another class of algorithms studied in the context of the minimum Steiner tree problem. This approach has been improved in a series of papers. The algorithm due to Robins and Zelikovsky (2005) currently reaches the best approximation ratio of 1.55. Byrka et al. (2010) proposed a randomized technique that achieves an approximation ratio of $ln(4) + \epsilon$, i.e. 1.39 in the limit. While the algorithm can be derandomized to obtain a deterministic approximation algorithm with polynomial time complexity, the polynomial and constants required to reach the approximation factor of 1.39 result in a runtime which is not feasible in practice. In our analysis, we start with the best known approximation algorithm by Robins and Zelikovsky (2005), before we analyze the approach by Mehlhorn (1988), and primal-dual algorithms (Goemans and Williamson 1995). These are arguably the most prominent approaches to the minimum Steiner tree problem in the literature.

We focus on the design of approximation mechanisms, i.e. approximation algorithms that can be implemented in dominant strategies. The field of algorithmic mechanism design has made substantial progress in the past years, and there are general frameworks to achieve truthfulness with randomized approximation mechanisms, and deterministic approximation mechanisms for specific problems. For example, a well-known black-box method to convert approximation algorithms for any packing problem into strategyproof mechanisms is the framework by Lavi and Swamy (2011), which is a randomized approximation algorithm.

Yet randomized approximation algorithms are often not acceptable in industrial procurement. Unfortunately, as of now there is no general framework to transform deterministic approximation algorithms into strategyproof mechanisms. However, there exist quite general approaches when additional conditions on bidders' valuations are met. Single-mindedness has received most attention in the literature on combinatorial auctions (Lehmann et al. 2002). It means that bidders are only interested in one specific subset of items (package). This can be a reasonable assumption for many real-world markets, and it is a very good starting point for our analysis of strategyproof approximation mechanism for the minimum Steiner tree problem on graphs. In the context of network procurement, we talk about bidders with single-dimensional types, which means each supplier only having access to a single link which she can sell.

Mu'alem and Nisan (2008) extended the framework of Lehmann et al. (2002) and presented conditions for approximately efficient and strategyproof mechanisms and single-minded bidders. Apart from this, numerous approximation mechanisms have been developed for specific algorithmic problems such as parallel scheduling and maximum flow problems (Archer and Tardos 2001), or graph traversal problems (Bilò et al. 2007). Interestingly, in spite of the importance of the minimum

Steiner tree problem, it has received very little attention in the literature on algorithmic mechanism design so far, with the only prior work being due to Gualà and Proietti (2005). They present a distance-network-based approximation mechanism which draws on the ideas of Takahashi and Matsuyama (1980).

We are particularly interested in the new class of deferred-acceptance auctions, which were introduced by Milgrom and Segal (2019) in the context of the Incentive Auction design for the US Federal Communications Commission (Leyton-Brown et al. 2017). Little is known so far about the solution quality deferred-acceptance auctions as compared to other deterministic and strategyproof approximation mechanisms in general. Dütting et al. (2017) is an exception, and they discuss approximation ratios of deferred-acceptance auctions for knapsack auctions as well as general combinatorial auctions with single-minded bidders. Recently, deferred-acceptance auctions have been generalized by Gkatzelis et al. (2017) for non-binary settings in which bidders do not simply win or lose but receive some level of service (e.g. a number of items awarded in a multi-item auction).

3 Notation and definitions

Let $G = (V, E, c)$ be a weighted, connected graph, where c_e is the cost of each edge $e \in E$. For a subset of edges $F \subseteq E$, the cost of the edge-induced subgraph is defined by $c(F) = \sum_{e \in F} c_e$. A *spanning tree* of G is a subset of edges of E such that the resulting edge-induced subgraph is connected, cycle-free and contains all vertices V. The *minimum spanning tree*, denoted by $MST(G)$, is a spanning tree where the sum of the costs of its edges is minimal in comparison with all other spanning trees.

The minimum Steiner tree problem on a connected graph $G = (V, E, c)$ is defined as follows. For a subset of vertices $K \subseteq V$ called *terminals*, any tree spanning K is called a *Steiner tree*. Any vertex in a Steiner tree which is not a terminal is called a *Steiner point*. We refer to the set of all Steiner trees over G as $StT(V, E)$. The objective then is to find a minimum cost Steiner tree.

Let G_V be the complete graph induced by the vertex set V, i.e., a complete weighted graph $G_V = (V, E_V, c_V)$, where each edge cost equals the cost of the shortest path in G between the two adjacent vertices of that edge. G_V is then a metric graph satisfying the triangle inequality. We call G_V the *distance network* of the graph G. Likewise, G_K denotes the distance network induced by the terminal set K, $G_K = (K, E_K, c_K)$. Note that $G_K \subseteq G_V$, as $K \subseteq V$.

In the following, we describe the design of mechanisms for the minimum Steiner tree problem. We consider a set of bidders N, where bidders $i \in N$ have single-dimensional types, i.e. each bidder i only provides one specific single edge e_i. With slight abuse of notation, we denote with c_i the true cost of bidder i while c refers to the corresponding tuple $(c_i)_{i \in N}$ taken over all bidders. Denote with B_i the domain of bids, i can report as her cost for edge e_i, e.g. $B_i = R_{\geq 0}$. B is defined as the Cartesian product $\prod_{i \in N} B_i$. For a *single-dimensional* bidder i, there is a unique and publicly known edge $e_i \in E$ such that her true private cost is c_i only for edge e_i, while for all other edges $e_j \neq e_i$ her true private cost is ∞. Given a vector of reported bids $b \in B$ with $b = (b_i)$, the expression b_{-i} denotes the bid tuple without the i-th entry, $b_{-i} = (b_j)_{j \in E \setminus \{i\}}$, and (c_i, b_{-i}) denotes the bid tuple where the i-th entry of b is replaced by c_i, i.e., bidder i reports her true cost.

A deterministic mechanism $\mathcal{M} = (f, p)$ for the minimum Steiner tree problem over vertices V and edges E is defined by a deterministic allocation function $f : B \rightarrow StT(V, E)$ and a payment scheme $p_i : B \times StT(V, E) \rightarrow \mathbb{R}$ for each bidder i. Given the bidders' reported bids $b \in C$, the mechanism $\mathcal{M} = (f, p)$ computes a Steiner tree $f(b)$ and pays each bidder i a payment of $p_i(b, f(b))$. In an approximation mechanism, the allocation function f is implemented via a deterministic approximation allocation algorithm $\mathcal{A}$. A mechanism with an approximation allocation algorithm $\mathcal{A}$ achieves an approximation ratio of r for minimum Steiner tree if

$$\max_{b \in B} \frac{c(OPT(b))}{c(\mathcal{A}(b))} \leq r$$

where $OPT(b)$ denotes a welfare-maximizing allocation (i.e. an optimal minimum Steiner tree given costs b), $c(OPT(b))$ the corresponding social welfare (i.e. cost of the Steiner tree), and $c(\mathcal{A}(b))$ the welfare achieved with the approximation algorithm $\mathcal{A}$.

Since bidders are self-interested, their reported bids b do not necessarily reflect their true costs c. Instead, bidders try to maximize their quasi-linear utilities u_i, i.e., payment received minus true cost: $u_i(b) = p_i(b, f(b)) - c_i$. As a result, a strategy-proof mechanism must offer bidders some incentives to reveal their true costs.

Definition 1 (*Strategyproofness*) A mechanism $\mathcal{M} = (f, p)$ is strategyproof if for all bidders $i \in E$ and all reported bid tuples $b \in B$ it holds that bidder i has a weakly higher payoff by telling the truth:

$$u_i(c_i, b_{-i}) \geq u_i(b)$$

Then, a bidder cannot make herself better off by not telling the truth about her costs. We also consider the stronger criterion of weak group-strategyproofness, where groups of bidders cannot make themselves better off by colluding.

Definition 2 (*Weak Group-Strategyproofness*) A mechanism $\mathcal{M} = (f, p)$ is weakly group-strategyproof if for every set of bidders $I \subseteq E$ and all reported bid tuples $b \in B$ it holds that at least one bidder $i \in E$ has a weakly higher payoff by telling the truth:

$$u_i(c_I, b_{-I}) \geq u_i(b)$$

In other words, in a weakly group-strategyproof mechanism it is impossible for a group of bidders to find alternative (non-truthful) bids that make all members of the group strictly better off.

We assume w.l.o.g. that for any two bidders i, j with $i \neq j$, it is $e_i \neq e_j$. If there are multiple bidders providing the same edge, we only consider the lowest reported bid for the allocation algorithm (though, of course, we consider all bids for the payment scheme). So from now on, we assume $i \widehat{=} e_i$. To avoid monopoly, we restrict G to be 2-edge-connected, i.e., G remains connected even if any single edge is removed.

With this, we can now formulate the minimum Steiner tree problem as a mechanism design problem: Let $G = (V, E, b)$ be a 2-edge-connected graph. $|V|$ is the number of vertices, $|E|$ is the number of edges/bidders, and b is the vector of reported bid prices. Let $K \subseteq V$ be the set of terminals. Then, the objective is to design a polynomial time approximation mechanism which computes an approximately efficient allocation A, and a payment scheme p which makes truthful bidding a dominant strategy, such that p and A form a strategyproof mechanism.

Definition 3 (*Monotonic allocation rule*) An allocation rule f of a mechanism $\mathcal{M} = (f, p)$ is monotonic if a bidder i who wins with bid b_i keeps winning for any lower bid $b'_i < b_i$ (for any fixed settings of the other bids).

Definition 4 (*Critical payment scheme*) A payment scheme p of a mechanism $\mathcal{M} = (f, p)$ is critical if a winning bidder i receives payment p^*_i, which is her maximum bid allowed for winning: $p^*_i := \sup\{b'_i \in B_i : i \in A(b'_i, b_{-i})\}$, where $A(b'_i, b_{-i})$ denotes the set of bidders that would have won if the reported bids were (b'_i, b_{-i})

Intuitively, a monotonic allocation ensures that a winner remains winning with any better bid, while the critical payment for a winning bidder is the highest cost that she may declare and still win. In his seminal paper, Myerson (1981) showed that an allocation rule f is implementable (i.e., there exists a payment vector p such that $\mathcal{M} = (f, p)$ is strategyproof) if and only if the allocation rule is monotonic. Moreover, if the allocation rule is monotonic and losing bidders pay 0, a critical payment scheme is the unique payment rule p such that $\mathcal{M} = (f, p)$ is strategyproof. Hence, with single-dimensional types and monotonic approximation algorithms, we can implement an outcome in dominant strategies, if we compute critical payments.

4 Approximation mechanisms for single-dimensional bidders

In this section, we briefly introduce important approximation algorithms for the minimum Steiner tree problem. A more extensive discussion can be found in the appendix. For the algorithms which can be extended to approximation mechanisms, we provide a corresponding critical payment scheme in Sect. 4.2, which then yields a strategyproof approximation mechanism. Finally, in Sect. 5 we design a deferred-acceptance auction for the minimum Steiner tree problem and discuss the worst-case approximation ratio of the deferred-acceptance auction and general greedy algorithms.

4.1 Approximation algorithms for the Steiner minimum tree

Table 1 lists the approximation algorithms for the minimum Steiner tree that we compare to the greedy approximation algorithms used in the deferred-acceptance auctions. These algorithms are representatives of very different approaches to

Table 1 Selected approximation algorithms for the minimum Steiner tree problem on graphs

Year	Approx. ratio	Authors	Monotonic?	Paradigm
1988	2.00	Mehlhorn	Yes	Distance network
1997	2.00	Goemans, Williamson	Yes	Primal-dual
2005	1.55	Robins, Zelikovsky	No	Loss-contracting

approximation. While two of them are monotonic, the one by Robins and Zelikovsky (2005) with the best approximation ratio so far is not.

We only provide an overview with a classification of whether these algorithms are monotonic or not. While this is fairly straightforward to answer for most approximation algorithms in the literature, the currently best class of algorithms for the minimum Steiner tree problem, the loss-contraction algorithms, are challenging to analyze. In the appendix we provide a detailed description of these algorithms and proofs of their monotonicity.

4.1.1 Loss-contracting algorithms

Loss-contracting algorithms have been the most successful approach to the design of approximation algorithms for the minimum Steiner tree on graphs so far. Any Steiner tree $S(G, K)$ of G is either a full Steiner tree, i.e., all its terminals are leaves, or can be decomposed into a forest of full Steiner subtrees (full components) by splitting all the non-leaf terminals (splitting a terminal results in two copies of the same terminal). The algorithm by Robins and Zelikovsky (2005) builds an *MST* on the subgraph G_K induced by the terminal set K and repeatedly adds full components to improve the temporary solution. In each iteration, full components are ranked according to their gain (by how much the component improves the current temporary solution) divided by their loss (i.e., the cost committed by adding a component or more precisely its Steiner points). After a full component is added, the temporary solution is improved. This step also involves loss-contracting, a method to make components which are in conflict with added ones less appealing. By these means, the algorithm by Robins and Zelikovsky (2005) achieves an approximation ratio of 1.55 if $k \to \infty$ and it is computable in $\mathcal{O}(|K|^k \cdot |V - K|^{k-2} + k \cdot |K|^{2k+1} \log |K|)$. This is the best approximation algorithm so far, but unfortunately it is not monotonic.

Proposition 1 *Allocation algorithm A^{RZ} is not monotonic.*

4.1.2 Distance-network-based approximations

Similarly to the loss-contracting approximation, the general idea of distance-network-based approximation algorithms is to build a *MST* on a complete subgraph $\overline{G_K}$ in the first phase. In the second phase, edges in $MST(\overline{G})$ are re-transformed back to edges in G, and an *MST* is computed on the resulting graph to remove possible cycles. Finally, in the third phase, non-terminal leaves are deleted. This algorithm was proposed by Kou et al and runs in $\mathcal{O}(|K||V|^2)$. However, due to the cycles that can occur in the first phase, this standard variant is not monotonic. Mehlhorn (1988) designed an algorithm which differs in phase 1. Here, the algorithm first partitions G into Voronoi regions, which are then utilized to construct a subgraph of G_K, called $\overline{G}$. It then proceeds with phase 2 and phase 3. This leads to a worst case runtime of $\mathcal{O}(|V|\log|V| + |E|)$ and achieves an approximation ratio of $2(1-1/l)$ where l is the minimal number of leaves in any minimum Steiner tree (which is naturally bounded above by the number of terminals). This algorithm is monotonic.

Proposition 2 *The allocation algorithm A^{MH} by Mehlhorn is monotonic.*

4.1.3 Primal-dual approximation algorithms

The approximation algorithm for the minimum Steiner tree problem by Goemans and Williamson (1995) follows a primal-dual approach in which the minimum Steiner tree problem is transformed into a hitting set problem and modeled as an integer linear program (IP). By relaxing the IP and considering its dual, Goemans and Williamson (1995) where able to propose an approximation algorithm that requires a runtime of $O(|V|^2 \log|V|)$ and also has an approximation ratio of 2.

Proposition 3 *The allocation of the primal-dual-based* minimum Steiner tree *approximation A^{PD} is monotonic.*

4.2 Payment schemes

Since the allocations of the algorithms A^{MH} by Mehlhorn (1988) and the primal-dual algorithm A^{PD} by Goemans and Williamson (1995) are monotonic, they meet the first requirement to be extendable to a strategyproof approximation mechanism. For the second requirement, the payment scheme p needs to find the critical payment p_i^* for any winner i, such that every reported bid b_i with $b_i \leq p_i^*$ is guaranteed to win, while every reported bid b_i with $b_i > b_i^*$ is guaranteed to lose.

We first discuss a payment scheme for distance-network-based approximation algorithms initially introduced by Gualà and Proietti (2005) which can be computed in $\mathcal{O}((|V| + |K|^2)|E| \cdot log\ \alpha(|E|, |V|))$. For this, consider the basic structure of an algorithm based on the distance network. Each winning edge e lies on at least one path connecting two terminals (v_1, v_2). If we now increase the cost of e, there are two possible causes that lead to e getting excluded from the solution. Either, there might be a shorter path between v_1 and v_2 that e is not part of. Apart from this, even if e is still on the shortest path between v_1 and v_2, the edge (v_1, v_2) might be replaced by some other edge (v_1', v_2') in the *MST* of distance network for which e is not on the resulting shortest path. The critical payment for e is then calculated by adding the difference between the original cost of the shortest path including e and the minimum cost of one of these alternatives without e.

The corresponding values can be calculated similar to (Gualà and Proietti 2005): First, the all-to-all distances problem is solved. Then, for every winning edge e on a shortest path between terminals v_1, v_2, an alternative shortest path between v_1 and v_2 that does not contain e can efficiently be computed using several tweaks (Buchsbaum et al. 1998; Gualà and Proietti 2005; Pettie and Ramachandran 2002). Computation of an alternative path in the distance network between two different terminals v'_1, v'_2 can also be done efficiently by standard sensitivity analysis (Tarjan 1982).

While the previous approach can be used for distance-network-based approximation, there is no efficient scheme for calculating the payments for the primal-dual approach and the loss-contraction algorithm by Robins and Zelikovsky (2005). In this case, another possibility to obtain critical payments is to find them through binary search. For a winning bidder i, a starting interval $[b_i, sp(b_{-i})]$, and first provisional payment $p^i_0 := \left\lfloor \frac{b_i+sp(b_{-i})}{2} \right\rfloor$, the binary search recursively computes a sequence of payments (p^i_j):

$$p^i_j = \begin{cases} \left\lfloor \frac{1}{2} p^i_{j-1} \right\rfloor & \text{if} \quad e_i \notin \mathcal{A}(G_{p^i_{j-1}}, K) \\ \left\lfloor \frac{3}{2} p^i_{j-1} \right\rfloor & \text{if} \quad e_i \in \mathcal{A}(G_{p^i_{j-1}}, K) \end{cases}$$

where $G_{p^i_{j-1}} = (V, E, (p^i_{j-1}, b_{-i}))$ is the original graph G with the only change being that the reported bid price of bidder i for his edge e_i has now become p^i_{j-1} instead of b_i, and $\mathcal{A}$ is the corresponding approximation algorithm. So $e_i \in \mathcal{A}(G_{p^i_{j-1}}, K)$ means that reporting p^i_{j-1}, bidder i still wins with all other bids fixed. This means that each computed payment p^i_j is tested by the respective allocation algorithm.

Since the payments described above are critical, they can be used in combination with their corresponding approximation algorithm to yield a strategy-proof approximation mechanism for single-dimensional bidders due to the results of Nisan et al. (2007).

5 Deferred-acceptance auctions

Greedy algorithms are an important class of approximation algorithms. A greedy-in algorithm iteratively chooses the best available option based on the current state (i.e., the previous iterations) and adds it to the solution. Contrarily, in the deferred-acceptance auction (DAA), a greedy-out algorithm is used which removes the least favorable alternative from the solution in every iteration. A greedy-in procedure which greedily accepts edges is not suitable for constructing a Steiner tree, since greedily accepting edges leads to being forced to 'correct' the structure afterwards (e.g. assuring that Steiner points do not end up as leaves), while within a greedy-out procedure one only needs to assure that it is still possible to construct a Steiner tree based on the remaining edges (i.e. all terminals are still connected). Thus, this section describes a greedy-out approximation for the Steiner tree problem implemented as a DAA (Milgrom and Segal 2019). The DAA is not only strategyproof but also weakly group-strategyproof and therefore provides a form of protection against bidder collusion.

The DAA greedily excludes the least desirable option from the solution until further removal would lead to an infeasible solution. To decide which option should be excluded in each iteration a scoring function is used. A scoring function assigns a value of at least 0 to an option i based on the cost of i and the remaining options. It is important to note that only the presence of remaining options, not their cost, may be considered in the scoring function as otherwise the mechanism might lose its incentive properties. Also, a scoring function needs to be non-decreasing in the first argument (cost of i). In each iteration, the option with the highest assigned score is removed from the allocation, options that cannot be removed without making the resulting solution infeasible receive a score of 0. All remaining edges with a score of 0 are accepted in the end. Hence, the algorithm always returns a feasible Steiner tree at the end. A representation of the algorithm is given below (Algorithm 1).

```
Data: 2-connected graph G = (V, E, b), terminal set K ⊆ V, a scoring
      function s
Result: A Steiner tree in G spanning K.
while true do
    for each edge e do
        assign score s(e) to e
        if s(e) = 0 then
            compute payment p(e)
        end
    end
    if highest score equals 0 then
        return remaining edges (Steiner tree)
    end
    remove e with highest score
end
```

Algorithm 1: Deferred-acceptance auction (DAA)

The payment $p(e)$ for an alternative e is calculated the moment we cannot exclude e from the solution any more, i.e. the moment we assign a score of 0 to it. The payment is equal to the bid e could have stated such that her score would have been equal to the one removed in the last iteration.

In a network procurement context, the set of options is the set of edges E. An edge cannot be excluded from the solution if its removal would lead G to decay into two connected components each of which contains at least one terminal. To account for the specific requirements of the network procurement context, we analyze three scoring functions in our experimental analysis:

1. the weight of the edge

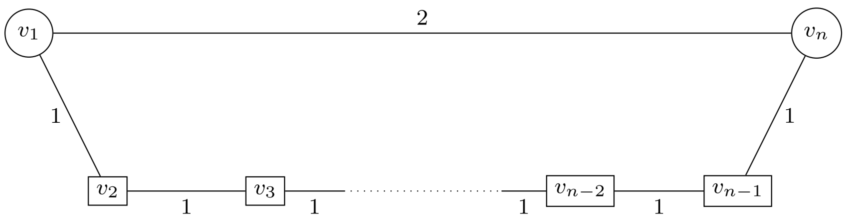


Fig. 1 An example where DAA leads to an approximation ratio of $(n-1)/2$

2. the weight of the edge divided by the number of adjacent edges
3. the weight of the edge divided by the betweenness centrality of the edge, where the betweenness centrality for each edge is defined by the number of shortest paths within the graph that use this edge

We calculate betweenness centrality by using the algorithm by Kourtellis et al. (Kourtellis et al. 2015) on a variant G_u of G where all edges have weight 1, i.e. G_u is the unweighted version of G. This is necessary, since due to incentive reasons a scoring function may only take the respective bid and the underlying graph structure into account, not the bids of other active bidders. Since in our environment bids are the cost of edges, when calculating the score of an edge e, we must ignore the costs of all other edges $e' \neq e$ in our calculations. In the following, let DAA_w (DAA_a;DAA_c) denote the DAA with scoring by weight (divided by adjacent edges; betweenness centrality).

Proposition 4 *The DAA for the* minimum Steiner tree *problem runs in* $\mathcal{O}(|E|^2 + |E||V| + t)$ *including payment calculation where t is the time necessary to update the scores.*

Proof In each iteration, at most $|E|$ scores need to be updated. This takes $|E| + |V|$ operations once to check for connectivity by Tarjan's bridge finding algorithm (Tarjan 1974) and constant time to update the score. Since there are at most $|E|$ iterations, this leads to a total runtime of $\mathcal{O}(|E|^2 + |E| \cdot (|E| + |V|)) \subseteq \mathcal{O}(|E|^2 + |E||V|)$ for DAA_w and DAA_a. For DAA_c betweenness centrality has to be calculated. This can be done in $\mathcal{O}(|E||V|)$ for each removal, i.e. in each iteration, using Kourtellis' algorithm (Kourtellis et al. 2015). Hence, the total runtime for DAA_c is $\mathcal{O}(|E|^2 + |E||V| + |E|^2|V|) \subseteq \mathcal{O}(|E|^2|V|)$. In all variants, payments need to be calculated for at most $|E|$ edges in constant time each. Thus, the runtime complexity is dominated by score updates. □

Greedy algorithms are usually fast, but can lead to arbitrarily bad results compared to an optimal solution for some problems. For instance, consider the three weight functions discussed above and a network as shown in Fig. 1. The network consists of n nodes $v_1, \dots, v_n$, two of which (v_1 and v_n) are terminals. The optimal Steiner tree consists of only keeping edge (v_1, v_n), but under DAA this edge is rejected first under all weight functions, forcing the algorithm to accept all other $n-1$ edges in order to remain connected. This leads to an approximation ratio of

$(n-1)/2$ proving the impossibility of a constant-factor approximation ratio. It remains an open question whether there exists a weight function that allows for such an approximation ratio.

6 Experimental evaluation

In the following, we present a thorough analysis of the different mechanisms discussed in this paper. For the approximation mechanism based on Mehlhorn (1988) and the primal-dual algorithm of Goemans and Williamson (1995), we computed the payments as described in Sect. 4.2. For the former, we employed the payment scheme for distance-network-based approximation algorithms by Gualà Proietti, and for the latter, we calculated the payments based on binary search. For the DAA, we use the threshold payments which are dynamically updated throughout the run of the algorithm as described in Sect. 5. Finally, we also included the Vickrey–Clarke–Groves mechanism as a baseline. We used the send-and-split method (Erickson et al. 1987) as implemented by Iwata and Shigemura (2018) to determine optimal solutions. All algorithms were implemented in Java. The experiments were executed on a laptop with Intel core i5-6600k (4 cores, 3.5 GHz) and 8GB RAM. We first describe the data set in Sect. 6.1 before we presenting our results in Sect. 6.2.

6.1 Data

Experiments are conducted on set I080 of the SteinLib Testdata Library (Koch et al. 2000).[1] Instances which are not 2-edge-connected are not considered since a monopoly edge would be worth infinite amounts of money. Thus, instances with names ending on 0x or 3x are not included. The remaining instances covered graphs with 4, 8, 16, and 20 terminals and densities of 11, 20, and 100 percent (very sparse as well as complete graphs). For each combination of terminal and density, the SteinLib test set contained 5 instances, i.e. a total of 60 instances. In order to investigate the effect of a graph's density on the performance of algorithms, we created additional instances with more diverse density values. Based on complete instances in I080, we created instances with densities between 0.3 and 0.9 (in increments of 0.1) by deleting edges randomly. For each combination of number of terminals and density, we generated 5 instances, for a total of 140 additional instances. Overall, we have an extensive set of 200 instances and 6 algorithms, resulting in 1200 experiments.

6.2 Results

Let us now summarize the results with respect to allocative efficiency, runtime, and revenue. We conclude this section with a short discussion of the results, comparing the different mechanisms.

[1] http://steinlib.zib.de/showset.php?I080.

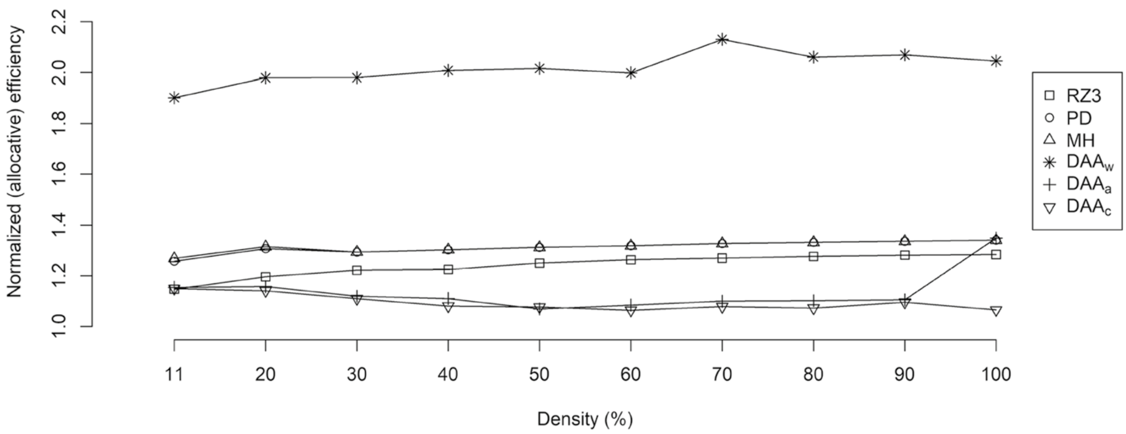


Fig. 2 Average efficiency over all instances

6.2.1 Allocative efficiency

In Fig. 2, we illustrate the efficiency of the five algorithms considered in our experimental evaluation for different levels of density. For each level of density, we show the mean efficiency of the algorithm for 20 instances (5 instances each for 6, 8, 16, and 20 terminals). While the approximation algorithm by Robins and Zelikovsky (2005) is not monotonic and thus cannot be extended to an approximation mechanism, it is still interesting to compare its allocation efficiency to the other algorithms in a complete information setting. Overall, DAA_c and DAA_a were the best performing algorithms and the scoring function based on the betweenness centrality came out to be the best scoring function for DAAs.

With a paired Wilcoxon rank-sum test, the differences in efficiency between *MH* and *PD* ($p = 0.059$) were not significant at $p < 0.0001$, while all other pairwise comparisons were significant at this level. We also analyze differences in efficiency using a linear regression with efficiency as dependent variable, the algorithm, the number of edges and terminals as covariates. With the DAA_c as baseline, the differences to this greedy algorithm were positive and significant at the following levels: DAA_w ($p < 0.0001$), *MH* ($p < 0.0001$), *PD* ($p < 0.0001$), *RZ* ($p < 0.0001$), and DAA_a ($p < 0.01$). The estimated coefficients further allow us to order the algorithm with respect to efficiency. Since we used the DAA_c as the baseline and all estimated coefficients are positive, we see that this approach provides the best results. The DAA_a (coefficient: 0.04) and the algorithm by Robins and Zelikovsky (coefficient: 0.14) follow closely, while both Mehlhorn's algorithm and the primal-dual approach exhibit a coefficient of 0.22. Finally, the DAA_w has the highest coefficient of 0.92.

Let us now report averages for different subgroups of the experiments in more detail. The algorithm by Robins and Zelikovsky performs best for sparse instances, on average finding a solution only 25% worse than the optimum and even solutions as good as 1.01 times the optimum (instance I080-015 of SteinLib). Moreover, it performs well for complete graphs, too (1.3 approximation ratio). Mehlhorn's algorithm and the primal-dual algorithm achieve similar results (130–140% of the optimum). The performance of these approximation algorithms gets slightly worse the denser the graph is.

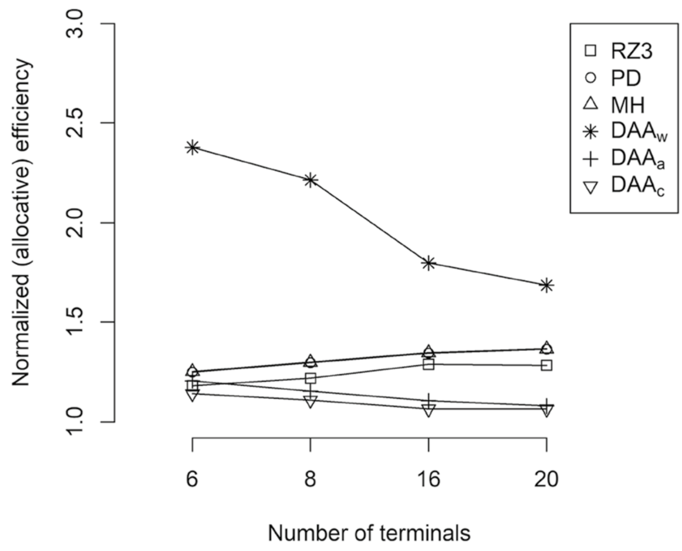
The performance of the DAA heavily depends on the scoring function. Using only the weight of an edge c_e as a score, allocative efficiency is never better than 1.48 times of the optimum, usually worse than 1.6 times of the optimum. It seems clear that without taking into account the structure of the graph, the greedy algorithm employed in DAA_w cannot compete with more sophisticated methods. Even in later stages of the DAA, edges are only selected based on their individual cost without considering the possible paths this edge is a part of. DAA_c (and especially, for smaller densities, DAA_a) generally provides better results on average than the more sophisticated approximation mechanisms. Only on very sparse instances, the algorithm by Robins and Zelikovsky can keep up with the performance of the DAA variants DAA_a and DAA_c. If we use edge weight divided by number of adjacent edges as scoring function, DAA_a performs better than the primal-dual algorithm for most sets of instances and even achieves results that are better than the results of the algorithm by Robins and Zelikovsky, except for the instances with 100 percent density.

The performance of DAA_a decreases significantly between a 90 percent density and 100 percent density. We generated further instances, increasing the density by a single percentage point between 90 and 100 percent. The efficiency ratio steadily increases between 90 and 100 percent. While DAA_a is equivalent to DAA_w in the first stages in very dense graphs (since every edge has the same number of adjacent edges), this effect does not come into play except for very dense graphs. Overall, DAA_a performs significantly better than DAA_w (on a significance level of 0.1%). Moreover, it can be seen that efficiency of DAA_a and DAA_c is nearly identical for sparse graphs and even up to a density level of 90 percent. In sparse graph, the number of possible paths between two nodes is more limited. Since an edge e with a lot of adjacent edges naturally allows for more paths (and hence also more shortest paths) to pass through e, the betweenness centrality of e is very dependent on its adjacent edges. Therefore, the DAAs with the corresponding scoring functions perform very similarly.

Figure 3 shows the average performance of the algorithms depending on the number of terminals (averaged over all densities, i.e. 50 instances per number of terminals). It can be seen that performance of all DAA variants improves as the number of terminals grows. This is to be expected since a greedy-out procedure actually solves MST optimally and the Steiner minimum tree problem becomes more like the MST problem for an increasing number of terminals (in the limit, when all vertices are terminal, they are identical). In contrast, all other approximation algorithms perform worse the more terminals are present in the graph.

In the following, we discuss the efficiency of the algorithms for fixed number of terminals. Since results for 6 and 8 as well as for 16 and 20 terminals, respectively, are very similar, we consider instances with a small number (6) and large number (20) of terminals. To improve readability, in the following we excluded DAA_w from the graphs and discussion as it was performing worse than all other mechanisms, in general. For 6 terminals, the density appears not to have a significant impact for *MH*, the primal-dual approach, and DAA_a (Fig. 4). This can easily be seen by means of linear regression where the p-value of density is > 0.1 for all of them. The algorithm by Robins and Zelikovsky performs worse the denser the graph is ($p < 0.001$), an effect easily observable in Fig. 4. In contrast, the DAA_c

Fig. 3 Average efficiency depending on number of terminals



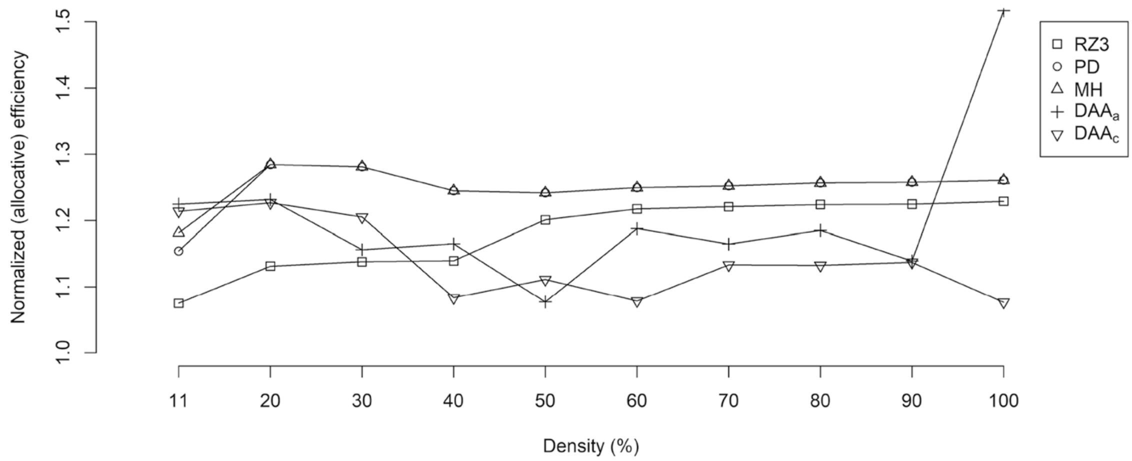


Fig. 4 Efficiency for 6 terminals

reveals favorable behavior if the graph is denser ($p < 0.01$). We can clearly see that independent of the number of terminals, DAA_a performs well for density levels up to 90 percent and then becomes significantly worse for very dense graphs.

This is in line with the results for 20 terminals where the density leads to better efficiency of the DAA_a for density values in [0.11, 0.9] ($p < 0.001$).

Figures 4 and 5 show that the *MH* and *PD* find very similar solutions that are below 1.4 times the optimal solution. The denser the graph, the worse the solutions these algorithms find. The algorithm by Robins and Zelikovsky finds even better solutions, but its performance decreases not only with increasing density but also as the number of terminals grows. Only for sparse instances with few terminals, it finds more efficient solutions than the more sophisticated DAA variants. DAA_a and the DAA_c find similarly efficient solutions for sparse graphs, where no significant difference was observed. The solutions found by the DAA_c are never worse than 1.09 times the optimum and in eight out of ten instances at most 6% worse than the optimal value for complete graphs with 16 or 20 terminals.

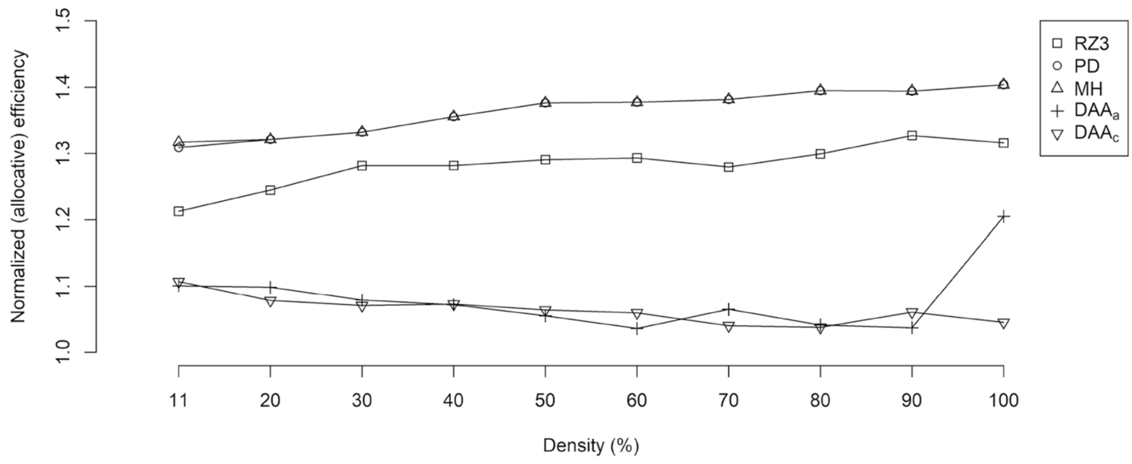


Fig. 5 Efficiency for 20 terminals

Table 2 Runtime combined means (in seconds)

Density	6 Terminals		20 Terminals	
	11%	100%	11%	100%
MH	0.24	0.46	0.33	1.26
PD	0.28	3.14	1.93	12.97
DAA_w	0.33	2.55	0.14	2.52
DAA_a	0.71	3.03	0.28	3.25
DAA_c	5.82	109.11	4.50	95.14
VCG	2.05	1.80	67544.54	58556.20

6.2.2 Runtime

In the following, we discuss the combined runtime required for the approximation mechanism to obtain both allocation and payments. Since the relative runtimes between mechanisms show a continuous pattern when incrementing the number of terminals and density, we only discuss extreme cases. Table 2 depicts the runtimes for the smallest and highest densities, as well as the smallest and largest numbers of terminals. Over all instances and densities, pairwise differences between two algorithms are significant at a level of $p < 0.0001$ using a Wilcoxon rank-sum test with the difference between the mechanism based on Mehlhorn's algorithm and the DAA_w being the only exception ($p = 0.026$). All mechanisms are significantly different from *VCG* ($p < 0.0001$). Our experiments show that *MH*, DAA_w, and DAA_A are the fastest group with *PD* following closely for instances with 6 or 8 terminals. Runtimes observed for the fastest group are lower than 4s on average on the set of instances with high numbers of both terminals and edges. Performance for lower numbers of terminals or edges is even better. Within the fastest group, the mechanism by Mehlhorn takes less time on complete graphs but is usually outperformed by the faster DAA variants for sparse graphs with more than 6 terminals.

Arguably, the more advanced payment computation used within *MH* leads to faster completion than calculating prices for *PD*, although we observed higher

Table 3 Revenue combined means (in arbitrary bid's currency)

Density	6 Terminals		20 Terminals	
	11%	100%	11%	100%
MH	2004.40	1496.00	6569.80	5580.00
PD	1920.40	1493.40	6654.80	5578.20
DAA_w	3376.60	2703.00	7393.60	6822.80
DAA_a	2152.27	2937.95	6133.26	9561.76
DAA_c	2735.85	2708.28	7404.98	6904.70
VCG	1644.80	1191.60	5078.80	4170.20

allocation runtime of the primal-dual algorithm when prices were not considered. The runtime required by all DAA variants is very dependent on the density of the graph, while it scales very well with the number of terminals. In our test instances, the computation time even slightly decreases with an increasing number of terminals in contrast to all other mechanisms. VCG in particular is very sensitive to higher number of terminals. Calculating exact solutions for minimum Steiner tree problems in the case of 20 terminals proved to be computationally inefficient, requiring a factor of up to 370.000 of the runtime as compared to the best performing DAA variants. Differences between the simpler scoring functions DAA_w and DAA_a are very small, while the calculation of betweenness centrality leads to higher runtimes for the DAA_c.

6.2.3 Revenue

Table 3 shows the extreme cases with the lowest density and highest density as well as the lowest and largest number of terminals. We have decided on this type of report, because again the developments between the extremes are smooth. In general, the payment increases with the number of terminals (since more edges need to be bought) and decreases with the density (since more competition between bidders allows for selecting cheaper options). The only exception is DAA_a with 100 percent density due to the bad (and hence more expensive) outcome that is obtained by the algorithm. Overall, the payments are lowest for VCG and the more sophisticated approximation algorithms yield a lower payment than the DAA variants (except for DAA_a for very sparse graphs).

The differences between DAA_w and DAA_c, *MH* and DAA_w, *MH* and DAA_c, *PD* and DAA_w, *PD* and DAA_c and *VCG* and all other algorithms are significant at a $p < 0.0001$ level using a Wilcoxon rank-sum test. The other pairs were not significantly different at this level, and their revenues are close. A regression with the sum of payments as the dependent variable and the number of edges, the number of terminals, and the algorithm as predictor variables shows that there are significant differences in revenue among most mechanisms. With *PD* as a baseline, we find that this approach yields lower payments than DAA_w ($p < 0.0001$), DAA_a ($p < 0.0001$), and DAA_a ($p < 0.0001$). No significant difference to the payments computed for the mechanism based on Mehlhorn's algorithm was found at this level. Using the *VCG*

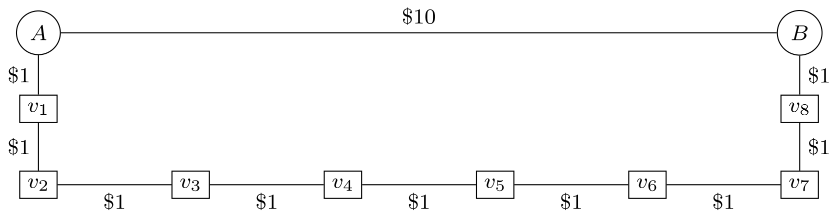


Fig. 6 An example where DAA leads to high payments

Table 4 Average seller utility (in % of their cost)

Density	6 Terminals		20 Terminals	
	11%	100%	11%	100%
MH	19.87	1.33	11.55	1.05
PD	15.15	1.25	12.64	1.02
DAA_w	2.65	1.69	2.71	1.45
DAA_a	22.46	70.41	22.01	106.54
DAA_c	46.22	103.66	48.08	65.70
VCG	13.65	1.85	12.51	6.00

mechanism as a baseline, all other algorithms yield significantly higher payments ($p < 0.0001$).

We also report results on the mean total utility calculated as the difference of total cost and total payments as percentage of total cost in Table 4. For *MH* and *PD*, the payments offered to winners of complete instances are low compared to their costs leading to low payoffs. For the DAA_a and DAA_c, sellers get a much higher payoff in all settings. The costs using the VCG mechanism are lowest.

Overall, we observe that the DAA mechanisms require significantly higher payments than VCG or the two approximation algorithms. While all of the mechanisms are strategy-proof, the allocation and payments are computed very differently. To see this, consider the example in Fig. 6 with two nodes *A* and *B* connected via a direct edge whose cost is \$10. There is another path between these two nodes with 9 edges with a cost of \$1 each. With threshold payments in a DAA, each of the winning edges gets \$10 when the direct edge is removed, and the overall revenue of the bidders on the 9 winning edges is \$90. With critical payments as they were used for MH and PD, the payments on each edge would be the maximum bid that would have still made the specific bidder winning (i.e., $2 - \varepsilon$), and the resulting revenue would be less than \$18.

7 Conclusions

In this paper, we showed which approximation algorithms can be extended to approximation mechanisms for the Steiner minimum tree problem with single-minded bidders. We proved that the best known algorithm by Robins and Zelikovsky violates monotonicity, a necessary condition for implementability. However, the algorithms by Mehlhorn (1988) and the primal-dual algorithm by Goemans and Williamson (1995) are monotonic and could be extended to strategyproof approximation mechanisms. Further, we designed a deferred-acceptance auction for the minimum Steiner tree problem and analyzed several scoring functions.

While the worst-case approximation ratio of deferred-acceptance auctions can be very low, the average-case solution quality is remarkably high, as shown in our numerical experiments. The results show that the group-strategyproof DAA with a scoring function based on betweenness centrality (DAA_c) yields very high allocative efficiency at low computation times in most environments (characterized by density and the number of terminals). DAA_a with a scoring function based on the number of adjacent edges performs similarly well (and even better for graphs with a higher number of terminals). Only for very dense graphs, the algorithm's solution quality declines.

Obviously, in terms of allocative efficiency, the exact *VCG* mechanism is best, but runtime can be prohibitive as the number of terminals grows. The runtime of all other algorithms is less than two minutes even for the large instances. *MH* exhibits a very low runtime over all instances and lower payments than DAA variants. Its allocative efficiency is significantly worse than that of DAA_a and DAA_c except for very sparse graphs with a low number of terminals. Overall, DAA_a has the best solution quality for density levels between 20 and 90 percent.

A number of questions are left for future research. The group-strategyproofness and the high average solution quality of the DAA variants come at the cost of higher payments for the buyer and higher profit for the sellers. This interesting trade-off for procurement managers requires further analysis to better understand which conditions justify the additional payment.

Another extension is to consider cases where bidders not only provide single edges, but packages of multiple edges. Unfortunately, this extension towards multi-dimensional mechanism design is far from trivial. One of the main problems is that bidders might not only lie about the cost of their edges but also about which edges they possess. This considerably complicates the construction of truthful mechanisms, such that we cannot rely on critical payments and monotonicity any more. In general, the design of deterministic approximation mechanisms for hard computational problems with multi-minded bidders remains a challenge.

Acknowledgements Open Access funding provided by Projekt DEAL. This project was supported by the Deutsche Forschungsgemeinschaft (DFG) (BI 1057/1-8) and (GRK 2201).

Appendix: Approximation algorithms for the minimum Steiner tree

In Section "Loss-contracting approximation: the algorithm by Robins and Zelikovsky," we discuss loss-contraction algorithms on the basis of the algorithm by Robins and Zelikovsky (2005), in Section "Distance-network-based approximations," we consider distance-network-based approaches, and in Sect. 1, the approximation by a primal-dual algorithm.

Loss-contracting approximation: the algorithm by Robins and Zelikovsky

Any Steiner tree $S(G, K)$ of G is either a full Steiner tree, i.e., all its terminals are leaves, or can be decomposed into a forest of full Steiner subtrees (full components) by splitting all the non-leaf terminals (splitting a terminal results in two copies of the same terminal). The algorithm builds a *MST* on the subgraph G_K induced by the terminal set K and repeatedly adds full components to the temporary solution. In each iteration, full components are ranked according to their gain (by how much the component improves the current temporary solution) divided by their loss (i.e. the cost committed by accepting a component or more precisely its Steiner points). After a full component is chosen, it is added to G_K. The full component is also added to the temporary solution in loss-contracted form. This ensures that components which are in conflict with accepted ones are less appealing subsequent iterations. Finally, a *MST* is built on the union of G_K and all chosen full components. By these means, the algorithm achieves an approximation ratio of 1.55 if $k \to \infty$ and is computable in $\mathcal{O}(|K|^k \cdot |V-K|^{k-2} + k \cdot |K|^{2k+1} \log |K|)$ (Robins and Zelikovsky 2005).

A k-restricted full component F is a full component with $k \geq 3$ terminals. By $C_l[F]$, we denote the loss-contracted full component of F. We define the gain and loss of a full component F formally and then describe the execution of Algorithm 2.

Definition 5 (*Gain and Loss of a Full Component* (Robins and Zelikovsky 2005)) Let T be a tree spanning K and F be an arbitrary full component of G given K.

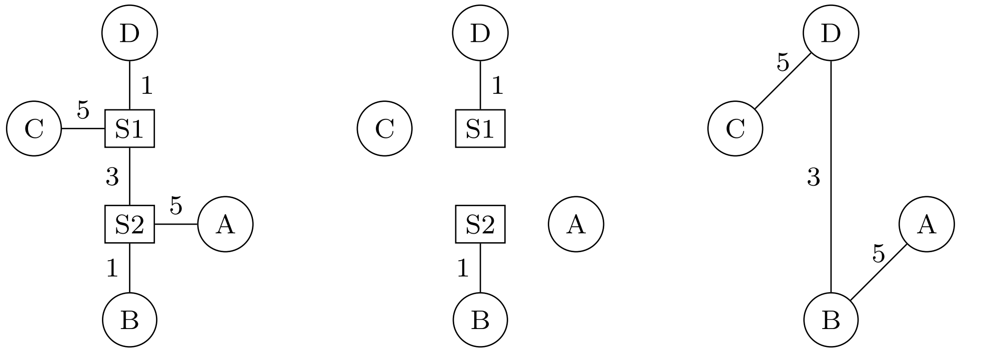
Let $T[F]$ be a minimum cost graph in $F \cup T$ which contains F completely and spans all terminals in K. This means $T[F]$ is the result of replacing a part of the tree T with the full component F.

Then, the Gain of F w.r.t. T is the cost difference between T and $T[F]$:

$$gain_T(F) = cost(T) - cost(T[F])$$

The loss of F is a minimum-cost subforest of F containing a path from each Steiner point in F to one of its terminals: $Loss(F_t) = MST\big(F_t \cup E_0(F_t)\big)\backslash E_0(F_t)$, where $E_0(F)$ denotes a complete graph containing all terminals of F, with all edge costs being 0. It follows that

$$loss(F) = cost\Big(MST\big(F \cup E_0(F)\big)\backslash E_0(F)\Big)$$



(a) Full component **(b)** Loss of full component **(c)** Contracted full component

Fig. 7 Full components

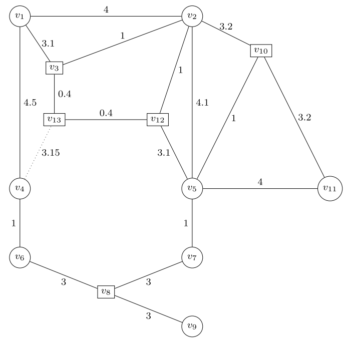
Data: 2-connected graph $G = (V, E, b)$, terminal set $K \subseteq V$, an integer k with $3 \leq k \leq |K|$
Result: A k-restricted Steiner tree $S(G)$ in G spanning K.
1 Compute G_V and G_K
2 $T = MST(G_K)$
3 **repeat**
4 Find a k-restricted full component F maximizing
5 $r = gain_T(F)/loss(F)$
6 $G_K = G_K \cup F$
7 $T = MST(T \cup C_l[F])$
8 **until** r $\leq$ 0;
9 $S(G, K) = MST(G_K)$
10 Replace artificial edges in $S(G, K)$
11 Cut Steiner point leaves of $S(G, K)$
12 **return** $S(G, K)$

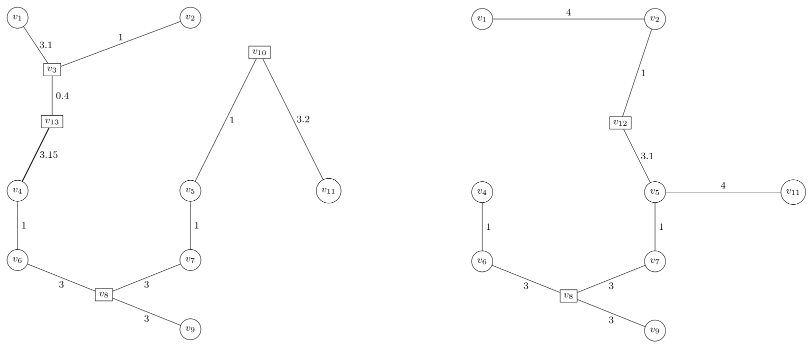
Algorithm 2: Approximation Allocation Algorithm A^{RZ}

The algorithm starts by computing G_V, its subgraph G_K (Line 1) and the *MST* on G_K (Line 2). Afterwards, the gain-over-loss ratios for all k-restricted full components are computed. It is sufficient to consider k-restricted full components consisting of k terminals and between 1 and $k-2$ Steiner points since every component is uniquely identified by its Steiner points of degree larger than 2 (Fig. 7a). Note that the gain of a full component F is dependent on T, while the loss is not. After choosing the full component with the highest gain-over-loss ratio, the selected component is added to G_K (Line 6). The component is also added to T in loss-contracted form $C_l[F]$ (Line 7).

To contract the loss of a full component F, we merge every connected tree of the forest $Loss(F)$ into a single vertex, the respective terminal of the component. Two terminals are connected in $C_l[F]$ if their respective components in $Loss(F)$ have an

Fig. 8 G





(a) solution for G where the owner of $\{V4, V13\}$ bids 3.15 **(b)** solution for G' where the owner of $\{V4, V13\}$ bids 3.11

Fig. 9 RZ: solutions **a** solution for G where the owner of $\{V4, V13\}$ bids 3.15 **b** solution for G' where the owner of $\{V4, V13\}$ bids 3.11

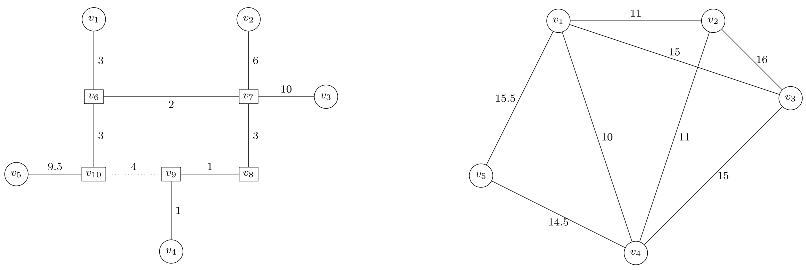
adjacent edge in F (Fig. 7b) and the cost of the edge in $C_l[F]$ is equal to the cost of the respective edge in F (Fig. 7c).

After $C_l[F]$ was added to T, an MST is built on $T \cup C_l[F]$. By improving T, the gain-over-loss ratio for the remaining full components is decreasing. Eventually, all components will have a gain-over-loss ratio of at most zero. At this point, the algorithm computes the $MST(G_K)$ (Line 9), transforms all its artificial edges back into original edges, i.e. replaces artificial edges by the respective shortest path, and cuts leaves which are Steiner points (Line 11).

Proposition 5 *Allocation algorithm A^{RZ} is not monotonic.*

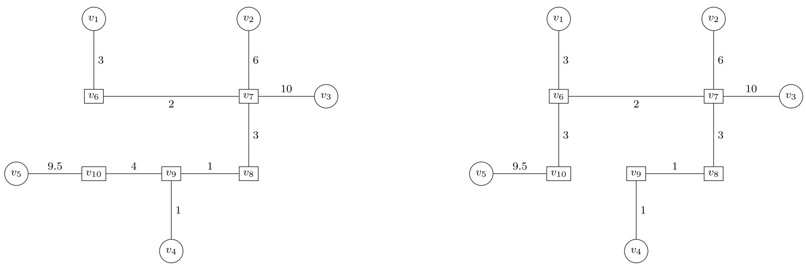
Proof The proof is by counter example. Consider graph G (Fig. 8) with terminal set $K = \{v_1, v_2, v_4, v_5, v_6, v_7, v_9, v_{11}\}$ (round nodes) and the non-terminals $\{v_3, v_8, v_{10}, v_{12}, v_{13}\}$ (rectangular nodes). The owner of $e = \{v_4, v_{13}\}$ bids 3.15.

The computed solution of cost 22.85 can be seen in Fig. 9a. Note that $e = \{v_4, v_{13}\}$ is part of the solution.



(a) G **(b)** relevant edges of G_K

Fig. 10 G and distance network G_K



(a) solution for G where the owner of $\{V9, V10\}$ bids 4 **(b)** solution for G where the owner of $\{v_9, v_{10}\}$ bids 3

Fig. 11 MST-based solutions. **a** Solution for G where the owner of $\{V9, V10\}$ bids 4 **b** solution for G where the owner of $\{v_9, v_{10}\}$ bids 3

However, if the owner of e reduces her bid to 3.11, the solution does not include e any more (Fig. 9b). Thus, monotonicity is violated and we cannot hope to achieve strategyproofness by using a critical payment scheme. □

So, the algorithm by Robins and Zelikovsky (2005) based on loss-contraction cannot easily be extended to a strategyproof mechanism. Let us next analyze approximation algorithms based on the distance network.

Distance-network-based approximations

Similarly to the loss-contracting approximation, the general idea of distance-network-based approximation algorithms is to build a *MST* on a complete subgraph G_K in the first phase. In the second phase, edges (shortest paths) in the *MST* are decomposed into edges in E, and a *MST* is computed on the resulting graph to remove possible cycles. Finally, in the third phase, non-terminal leaves are deleted. This algorithm was proposed by Kou et al. and runs in $\mathcal{O}(|K||V|^2)$. However, due to the cycles that can occur in the first phase, this standard variant is not monotonic. To see this, consider the graph G in Fig. 10 with its relevant edges of the respective distance network G_K. If the bid for $e = \{v_9, v_{10}\}$ is 4, e is part of the solution (Fig. 11a). However, if the bid was only 3, e might be removed from the solution (Fig. 11b). Hence, monotonicity is violated.

Gualà and Proietti (2005) change the algorithm in its second phase when the MST on the subgraph G_K is replaced by the corresponding shortest paths. Instead of adding all shortest paths and afterwards calculating the MST on the resulting graph to remove the cycles, in the extended algorithm the shortest paths are inserted iteratively in a way such that no cycles are introduced. The authors show that such an acyclic expansion is always possible. The resulting algorithm requires a runtime of $\mathcal{O}((|V| + |K|^2)|E| \cdot log\ \alpha(|E|, |V|))$ where $\alpha(.,.)$ is the classic inverse of the Ackermann's function as defined in Tarjan (1982) and yields a $2(1 - 1/|K|)$-approximation.

Mehlhorn (1988) designed an algorithm which differs in phase 1. Here, the algorithm first partitions G into Voronoi regions, which are then utilized to construct a subgraph of G_K, called $\overline{G}$. It then proceeds with phase 2 and phase 3 as described above. This leads to a worst case runtime of $\mathcal{O}(|V| \log |V| + |E|)$ and achieves an approximation ratio of $2(1 - 1/l)$ where l is the minimal number of leaves in any minimum Steiner tree (which is naturally bounded above by the number of terminals). In the following, we discuss Mehlhorn's algorithm and show that the allocation is monotonic. Hence, the algorithm is also suitable to be extended to an approximation mechanism with a slightly better runtime and approximation ratio than other algorithms based on distance-network-based approximation algorithms.

Definition 6 (*Voronoi Regions* $\mathcal{V}(s)$) Given a general graph $G = (V, E, b)$ and the set of terminals $K \subseteq V$, the *Voronoi region* $\mathcal{V}(s)$ of a terminal $s \in K$ contains all vertices $v \in V$ for which the shortest path $sp(s, v) \leq sp(t, v)$ for all $t \in K$. We break ties randomly, such that each vertex v uniquely belongs to one such region.

Definition 7 (*Distance network based on* $\mathcal{V}$) Let $\bar{G} = (K, E_{\bar{G}}, b_{\bar{G}})$ be the distance network with edges and weights as follows:

$$(s, t) \in E_{\bar{G}} \Leftrightarrow \exists\, (u, v) \in E \text{ such that } u \in \mathcal{V}(s) \text{ and } v \in \mathcal{V}(t)$$

$$b_{\bar{G}}(s, t) = \min\{sp(s, u) + b(u, v) + sp(v, t) \,:\, u \in \mathcal{V}(s), v \in \mathcal{V}(t), (u, v) \in E\}$$

Data: 2-connected graph $G = (V, E, b)$, terminal set $K \subseteq V$
Result: A Steiner tree $S(G, K)$ in G spanning K
1 Compute Voronoi regions of G and generate $\bar{G}$;
2 $S(G, K) = MST(\bar{G})$
3 Replace artificial edges in $S(G, K)$
4 Cut non-terminal leaves of $S(G, K)$
return $S(G, K)$

Algorithm 3: Approximation Allocation Algorithm $\mathcal{A}^{MH}$

Similar to the algorithm by Gualà and Proietti (2005), the algorithm by Mehlhorn is also monotonic.

Proposition 6 *The allocation of Mehlhorn's algorithm is monotonic.*

Proof Suppose there is a graph G, an allocation A computed by Mehlhorn's algorithm and an edge $e \in A$. Further assume that the owner of e lowers her bid. Reducing a bid for an edge e can only mean that e is now part of at least as many shortest paths as before. In general, this may change the allocation. However, only shortest paths which contain e are cheaper after the changed bid. Hence, e will remain part of the solution even though it might be chosen in another path. It remains to be shown that a changed allocation cannot lead to cycles in the solution and thus to the possible exclusion of e.

Suppose there is a cycle between two Voronoi regions. This would mean that two paths between the respective regions have been chosen. Since an *MST* is built on the subgraph induced by the Voronoi regions, this can never happen during Mehlhorn's algorithm. A similar argument holds for cycles in more than two Voronoi regions. Finally, there can be no cycles inside a single Voronoi region (by definition). Since no cycle can occur, no edge that has been added to the solution will be removed from the solution at a later point and therefore e is in the final solution. Thus, the allocation computed by Mehlhorn's algorithm is monotonic. □

Primal-dual approximation algorithms

This section describes the general approach for primal-dual approximations and the approximation algorithm for the minimum Steiner tree problem by Goemans and Williamson (1995) which requires a runtime of $O(|V|^2 \log |V|)$ and also has an approximation ratio of 2.

Many problems in graph theory can be reduced to the hitting set problem. For a ground-set E with cost $c_e \geq 0$ for every element $e \in E$ and subsets $T_1, T_2 \ldots T_n \subseteq E$, the hitting set problem is to find a subset $A \subseteq E$ of minimal cost such that $A \cap T_i \neq \emptyset$ for all subsets $i = \{1, \ldots n\}$. The primal integer program for the hitting set problem can be formulated as follows:

$$\begin{aligned} \text{Min} \quad & \sum_{e \in E} c_e x_e \\ \text{subject to} \quad & \sum_{e \in T_i} x_e \geq 1, \qquad \forall i \\ & x_e \in \{0, 1\}. \end{aligned}$$

To obtain the relaxation, simply the constraint $x_e \in \{0, 1\}$ needs to be relaxed to $x_e \geq 0$. The corresponding dual program is stated below:

$$\begin{aligned} \text{Max} \quad & \sum_i y_i \\ \text{subject to} \quad & \sum_{i:e \in T_i} y_i \le c_e, \; \forall e \in E \\ & y_i \ge 0, \quad \forall i. \end{aligned}$$

To obtain an α-approximation, we compute a solution $\bar{x}$ to the primal integer program and a solution y to the dual of the relaxed primal program such that $\sum_{e \in E} c_e \bar{x}_e \le \alpha \sum_{i=1}^{n} y_i$.

```
   Data: ground-set E, subsets T_1, T_2, ... T_n ⊆ E
   Result: allocation A
1  y = 0  ∀y
2  A = ∅
3  while A not feasible do
4  |   Find violated T_k  (T_k ∩ A = ∅)
5  |   Increase y_k until ∃e ∈ T_k s.t.  Σ_{i:e∈T_i} y_i = c_e
6  |   A = A ∪ {e}
7  end
8  return A
```

Algorithm 4: Approximation Algorithm for the hitting set problem

Algorithm 4 describes the necessary steps to compute A. During the initialization, A is empty and all dual variables y are set to 0. In each iteration, a violated set T_k is chosen. Afterwards, the corresponding dual variable y_k is increased (loaded) until one of the constraints holds with equality (it goes "tight," Line 5). The corresponding element e is then added to the solution. If the allocation A is feasible, the algorithm stops and returns A.

Mapping the hitting set problem to the minimum Steiner tree problem is straightforward: The ground-set is given by the edges E of the graph and c_e is the cost of the respective edge $e \in E$. Let S_i be a subset of vertices that contains at least one, but not all terminals, i.e. a cut. When all cuts are crossed, the solution is a feasible allocation for the minimum Steiner tree problem. By definition, the edges adjacent to exactly one vertex $v \in S_i$ are the edges crossing the cut S_i. Let $\delta(S_i)$ denote the set of these edges. Let $T_i = \delta(S_i)$. The adapted algorithm can be seen below (Algorithm 5). It achieves an approximation ratio of 2 (Goemans and Williamson 1995).

```
Data: 2-connected graph G = (V, E, b), terminal set K ⊆ V
Result: A Steiner tree S(G) in G spanning K
y = 0  ∀y
A_0 = ∅
i = 0
while A_i not feasible do
    Choose violated sets 𝒰
    Increase y_k uniformly for all T_k ∈ 𝒰 until ∃e_i ∉ A_i s.t. \sum_{i:e_i ∈ T_i} y_i = c_{e_i}
    A_i = A_i ∪ {e_i}
    i = i + 1
end
A' = A_{i-1}
for i; i ≥ 0; i = i − 1 do
    if A' \ {e_{t_i}} still feasible then
        A' = A' \ {e_{t_i}}
    end
end
```

Algorithm 5: Approximation Allocation Algorithm $\mathcal{A}^{PD}$

Two modifications can be seen in Algorithm 5 in comparison with the basic primal-dual algorithm (Algorithm 4). Firstly, load is not increased on one, but multiple (minimal) unsatisfied components $T_k \in \mathcal{U}$. $\mathcal{U}$ contains all T_k that are unsatisfied and minimal, i.e. there is no unsatisfied set T_j with $T_j \subset T_k$. Secondly, after computing the allocation A a reverse deletion is conducted. In this phase, edges are assessed in regard to their necessity in reversed order (LIFO). Unnecessary edges either connect a Steiner point as a leaf or close a cycle. In either case, the edge is not contributing to the solution (apart from inflicting costs).

Proposition 7 *The allocation of the primal-dual based* minimum Steiner tree *approximation is monotonic.*

Proof Suppose there is a graph G, an allocation A computed by the primal-dual approximation algorithm for Steiner trees and an edge $e \in A$ whose cost has been truthfully stated by its owner. Further assume that the owner of e lowers her bid to c'_e. Due to the lower cost, e can go tight only sooner. Since e was part of the first allocation, we know that conflicting edges have been removed before e was candidate for removal. Since e is now cheaper and was thus added to the solution earlier or at the same point, it still is considered for removal later than the conflicting edges. Hence, when e is assessed for necessity, the conflicting edges have already been removed. The allocation computed by the primal-dual approximation algorithm for Steiner trees is thus monotonic. □

References

Archer A, Tardos E (2001) Truthful mechanisms for one-parameter agents. In: Proceedings of the 42nd IEEE symposium on foundations of computer science. IEEE Computer Society, Washington, DC, USA, FOCS ’01, pp 482–491

Bilò D, Forlizzi L, Gualà L, Proietti G (2007) An algorithm composition scheme preserving monotonicity. In: Proceedings of the twenty-sixth annual ACM symposium on principles of distributed computing. ACM, New York, NY, USA, PODC ’07, pp 360–361

Buchsbaum AL, Kaplan H, Rogers A, Westbrook JR (1998) Linear-time pointer-machine algorithms for least common ancestors, MST verification, and dominators. In: Proceedings of the thirtieth annual ACM symposium on theory of computing, ACM, pp 279–288

Byrka J, Grandoni F, Rothvoß T, Sanità L (2010) An improved LP-based approximation for steiner tree. In: Proceedings of the forty-second ACM symposium on theory of computing. ACM, New York, NY, USA, STOC ’10, pp 583–592. https://doi.org/10.1145/1806689.1806769

Clarke E (1971) Multipart pricing of public goods. Public Choice XI:17–33

Contreras I, Fernández E (2012) General network design: a unified view of combined location and network design problems. Eur J Oper Res 219(3):680–697

Dütting P, Gkatzelis V, Roughgarden T (2017) The performance of deferred-acceptance auctions. Math Oper Res 42(4):897–914

Erickson RE, Monma CL, Veinott Jr AF (1987) Send-and-split method for minimum-concave-cost network flows. Math Oper Res 12(4):634–664

Gkatzelis V, Markakis E, Roughgarden T (2017) Deferred-acceptance auctions for multiple levels of service. In: Proceedings of the 2017 ACM conference on economics and computation. ACM, pp 21–38

Goemans MX, Williamson DP (1995) The primal-dual method for approximation algorithms and its application to network design problems. In: Hochbaum D (eds) Approximation algorithms for NP-hard problems. PWS Publishing, Boston, pp 144–191

Green J, Laffont JJ (1977) Characterization of satisfactory mechanisms for the revelation of preferences for public goods. Econometrica J Econ Soc 45(2):427–438

Groves T (1973) Incentives in teams. Econometrica 41:617–631

Gualà L, Proietti G (2005) A truthful (2 - 2/k)-approximation mechanism for the steiner tree problem with k terminals*. In: Proceedings of the 11th annual international conference on computing and combinatorics. Springer, Berlin, COCOON’05, pp 390–400. https://doi.org/10.1007/11533719_40

Iwata Y, Shigemura T (2018) Steiner tree solver [software]. https://github.com/wata-orz/steiner_tree. Accessed 8 May 2018

Karp R (1972) Reducibility among combinatorial problems. In: Miller R, Thatcher J (eds) Complexity of computer computations. Plenum Press, Berlin, pp 85–103

Koch T, Martin A, Voß S (2000) SteinLib: An updated library on steiner tree problems in graphs. Technical report ZIB-Report 00-37, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustr. 7, Berlin. http://elib.zib.de/steinlib. Accessed 12 Mar 2018

Kou L, Markowsky G, Berman L (1981) A fast algorithm for steiner trees. Acta Informatica 15(2):141–145. https://doi.org/10.1007/BF00288961

Kourtellis N, Morales GDF, Bonchi F (2015) Scalable online betweenness centrality in evolving graphs. IEEE Trans Knowl Data Eng 27(9):2494–2506

Lavi R, Swamy C (2011) Truthful and near-optimal mechanism design via linear programming. J ACM 58(6):25:1–25:24. https://doi.org/10.1145/2049697.2049699

Lehmann D, O’Callaghan LI, Shoham Y (2002) Truth revelation in approximately efficient combinatorial auctions. J ACM 49(5):577–602. https://doi.org/10.1145/585265.585266

Leyton-Brown K, Milgrom P, Segal I (2017) Economics and computer science of a radio spectrum reallocation. Proc Natl Acad Sci 114(28):7202–7209

Mehlhorn K (1988) A faster approximation algorithm for the steiner problem in graphs. Inf Process Lett 27(3):125–128

Milgrom P, Segal I (2019) Clock auctions and radio spectrum reallocation. J Polit Econ. https://doi.org/10.1086/704074

Mu’alem A, Nisan N (2008) Truthful approximation mechanisms for restricted combinatorial auctions. Games Econ Behav 64(2):612–631. https://doi.org/10.1016/j.geb.2007.12.009

Myerson RB (1981) Optimal auction design. Math Oper Res 6(1):58–73
Newman N, Leyton-Brown K, Milgrom P, Segal I (2017) Assessing economic outcomes in simulated reverse clock auctions for radio spectrum. CoRR abs/1706.04324. arxiv:1706.04324
Nisan N, Ronen A (1999) Algorithmic mechanism design (extended abstract). In: Proceedings of the thirty-first annual ACM symposium on theory of computing. ACM, New York, NY, USA, STOC '99, pp 129–140. https://doi.org/10.1145/301250.301287
Nisan N, Roughgarden T, Tardos E, Vazirani V (2007) Algorithmic game theory. Cambridge University Press, Cambridge
Öncan T, Cordeau JF, Laporte G (2008) A tabu search heuristic for the generalized minimum spanning tree problem. Eur J Oper Res 191(2):306–319
Pettie S, Ramachandran V (2002) Computing shortest paths with comparisons and additions. In: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, pp 267–276
Robins G, Zelikovsky A (2005) Tighter bounds for graph steiner tree approximation. SIAM J Discrete Math 19(1):122–134. https://doi.org/10.1137/S0895480101393155
Takahashi H, Matsuyama A (1980) An approximate solution for the steiner problem in graphs. MathJaponica 24:573–577
Tarjan RE (1974) A note on finding the bridges of a graph. Inf Process Lett 2:160–161
Tarjan RE (1982) Sensitivity analysis of minimum spanning trees and shortest path trees. Inf Process Lett 14(1):30–33
Vazirani VV (2013) Approximation algorithms. Springer, Berlin
Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. J Finance 16(1):8–37
Xu J, Chiu SY, Glover F (1995) Tabu search heuristics for designing a steiner tree based digital line network. University of Colorado, Boulder

Affiliations

Martin Bichler[1] · **Zhen Hao**[1] · **Richard Littmann**[1,2] · **Stefan Waldherr**[1]

Zhen Hao
zhen.hao@in.tum.de

Richard Littmann
richard.littmann@in.tum.de

Stefan Waldherr
stefan.waldherr@in.tum.de

[1] Technical University of Munich, Munich, Germany

[2] Advanced Optimization in a Networked Economy (AdONE), GRK 2201, Technical University of Munich, Leopoldstr. 145, 80804 Munich, Germany

4 Pricing with Budget Constraints

Journal Publication Status: Minor Revision, Transportation Research - Part B, 15.11.2020

Peer-Reviewed Conference Paper

Title: Combinatorial exchanges for airport time slots

Authors: R. Littmann, M. Bichler, S. Waldherr

In: Workshop on Information Technologies and Systems 2019

Abstract: Due to the lasting growth in air traffic, many international airports have reached their capacity limits. Access to major airports is granted through the assignment of airport time slots. Current practices of allocating these time slots via grandfathering are widely regarded as inefficient by experts. New market mechanisms need to take into account synergistic valuations of airlines for departure and arrival time slots as well as financial constraints of the participating airlines for the many time slots available. Unfortunately, computing core-stable outcomes in such environments is $\sum_2^{\mathrm{P}}$-hard. Such problems are typically considered intractable. We introduce bilevel integer optimization models for airport time slot trading and compute core-stable outcomes, i.e., allocations and prices such that no coalition can beneficially deviate. Interestingly, despite the computational hardness of the underlying problem, numerical experiments show that instances of practically relevant size can be solved in due time. The proposed market design provides a solution that addresses the specific constraints of airport time slot markets, a precondition for adoption in the field.

Contribution of thesis author: Results, implementation, mathematical model, presentation, joint project and paper management

Reference: Bichler et al. (2019)

Combinatorial exchanges for airport time slots

Complete Research Paper

Richard Littmann, Martin Bichler, Stefan Waldherr
Technische Universität München
{richard.littmann,bichler,stefan.waldherr}@in.tum.de

1 Introduction

Airport time slots for the start and landing of airplanes are a scarce resource at major airports. Nowadays, the assignment of slots by the airport coordinator is largely based on grandfathering. Only if a slot is not used up to 80% an airline loses this slot. Current practices have been challenged as inefficient. Airlines leverage grandfathering rights to block valuable slots for the competition, although there would be much more valuable uses (Avenali et al. 2015). It is widely accepted that "market mechanisms would improve allocative efficiency" by bringing appropriate incentives to bear to ensure that slots are allocated to whomever values them the most (Madas and Zografos 2010). Unfortunately, the design of such markets is considered very challenging, which is a major barrier to the adoption of market-based solutions. Jones et al. (2004) write:

> ··· auctions of 10% of slots, combined with secondary trading could, in theory, achieve the most efficient allocation of slots possible. But in practice, many of the auctions are likely to be so complex, both for auction organisers and for airlines bidding for slots, that it is probably unlikely that an efficient allocation of slots will emerge from this process.

This leads to interesting market design problems at the intersection of information systems and operations research, a field that has drawn considerable attention in the past decade (Adomavicius and Gupta 2005; Adomavicius et al. 2013; Bichler et al. 2010; Goetzendorff et al. 2015). This paper contributes to the theme of this year's workshop: markets for policy making and sustainability.

We propose a mechanism that improves the allocation of airport time slots while also taking into account important constraint the participants face, such as complementarities between take-off and landing slots, and budget constraints. We show that ignoring the latter leads to severe losses in welfare as well as instabilities when airlines are forced to submit bids lower than their actual values in order to avoid making a loss. These negative aspects are potentially very discouraging for airlines from participating in existing mechanisms for slot allocation. Stability is often seen as a first-order goal in market design (Roth 2002). The core is the most well-known notion of stability in game theory and it is natural to ask for core-stable outcomes of a market. Such an outcome requires that there cannot be any coalition of participants that have incentives to deviate from the allocation determined by the mechanism. It was shown that core-stability coincides with the notion of competitive equilibria in markets with payoff-maximizing bidders and that the core might be empty in a combinatorial exchange (Bikhchandani and Ostroy 2002).

Unfortunately, it turns out that the presence of budget constraints makes the computation of core-stable outcomes a much harder computational problem. The problem actually becomes Σ_2^p-hard, a complexity class that is one step above *NP* in the polynomial hierarchy (Bichler and Waldherr 2019). While problems in this complexity class are typically considered intractable, in this paper we design a market mechanism for slot allocation and show in experiments that problem instances of practically relevant size can be solved in reasonable computational run time.

We model the assignment and pricing of airport time slots as a bilevel optimization problem which takes into account the complementarities in take-off and landing slots as well as the financial constraints of airlines. Our mechanism then determines a stable allocation with the highest social welfare. Despite economic struggles of airlines, financial constraints have not yet been considered in the design of mechanisms for airport slot allocation. In a computational study, we demonstrate that ignoring budget constraints leads to severe welfare losses and instabilities when using current market mechanisms. In fact, our experiments show that gains from trade decrease by 15 to over 20% when airlines are forced to shade their bids in order to avoid making potential losses when participating in the auction. These results indicate that ignoring budget constraints may be a major factor for airlines to abstain from trading slots. While incorporating budget constraints leads to a very tough combinatorial optimization problem, we show that problem instances of relevant size can be solved.

In summary, our contribution is a mechanism that achieves four important goals: (i) Designing a central market for slot allocation, (ii) incorporating budget constraints, which are of high concerns in the airline industry, (iii) being efficient in finding stable outcomes, (iv) improving the welfare as opposed to mechanism that ignore budget constraints.

The paper is structured as follows. In Section 2, we describe the airport slot allocation problem and discuss related work. In Section 3, the mathematical model is presented and the mechanism to allocate airport time slots and derive prices is introduced in Section 4. We evaluate our mechanism in Section 5 before drawing our conclusions in Section 6

2 Problem Description

In this section, we briefly describe how airport slots are currently allocated and traded. We review a large number of papers that advocate for replacing the current process with central market mechanisms. Further, we discuss budget constraints and their effects on market mechanisms in general, as well as the prevalence of financial struggles in the airline industry.

2.1 Airport Slot Allocation Mechanisms

Congestion at airports has been a problem for many years and many attempts have been made to improve conditions. One of the most important improvements was the introduction of slot control systems in the 1960s. A slot at an airport is defined as a time interval available for the arrival or departure of a flight. Airports declare their capacity in number of available slots per hour, where the number of slots reflects the amount of air traffic an airport can handle within the specified time frame. The IATA (International Air Transport Association) slot guidelines were first issued in 1976. Nowadays, almost all European airports use slot allocation systems based on the IATA guidelines and the regulations of the European Commission. However, only four major US airports use slot control systems due to a variety of reasons (Ball et al. 2017).

Assignment and pricing mechanisms for slots have come under much scrutiny over the years. At participating airports, slots are allocated in two steps. In the *primary allocation*, which takes place twice a year, slots for the upcoming summer or winter season are allocated from airports to airlines. The number of available slots is based on IATA and European Commission guidelines and slots are allocated first according to grandfather rights, given that the requirements to keep these rights were reached. Afterwards, half of the remaining slots are distributed to new entrants. Only the remaining slots are open for sale and allocated to interested airlines. Grandfather rights are only lost in case that a slot is underutilized. Due to the limited number of slots, incumbent airlines will want to keep these slots simply to protect them from competition. Hence, airlines ”babysit” slots, using them for the minimum required amount in order to keep their grandfather rights. Ball et al. (2017) state that “It [is] clear that current slot allocation policies do not encourage allocating slots to their highest and best use, and in some respects actually discourage it”.

After the primary allocation, airlines can participate in *secondary trading* of slots during the season. While these secondary markets have been used in the US, they currently are not in place. On the other hand, secondary trading is allowed in Europe where it is legal to exchange slots, even with additional financial transactions based on a 1999 English High Court judgment in R v Airport Coordination (Ball et al. 2017). However, participation in these secondary markets is low. A 2006 report of the European Commission reported that only 1.2 % of slots were traded at London Heathrow, Europe’s busiest airport.

Ball et al. (2017) argue that nowadays there is a strong case for the use of market mechanisms. In fact, auction mechanisms have long been proposed for the primary allocation of airport time slots (Rassenti et al. 1982; Castelli et al. 2011; Pellegrini et al. 2012; Ball et al. 2006). Similarly, strong cases have been made for centralized mechanisms for secondary trading (Ltd. 2001; Pellegrini et al. 2012), as currently secondary trading only occurs through bilateral exchanges. Unfortunately, an abolishment (or severe limitation) of grandfather rights seems like a necessity for the successful implementation of mechanisms for primary markets. This may well be a reason for the backlash these proposals have received in the past. Hence, in this paper we will focus mostly on the design of a centralized market for secondary slot trading.

Package bids are essential in the slot allocation domain and need to be considered by any mechanism in order to produce stable outcomes. A takeoff slot at a flight originating airport is only valuable with a landing slot at the flight destination airport. Combinatorial auctions and exchanges allow participants to state these package bids. Since its inception (Rassenti et al. (1982), tellingly using airport slot auctions as a motivating example), combinatorial auctions have found widespread application for the sale of spectrum licenses (Bichler and Goeree 2017), day-ahead energy markets (Martin et al. 2014), and supply chain coordination (Fan et al. 2003; Guo et al. 2012; Walsh et al. 2000).

2.2 Budget Constraints

In order to result in efficient outcomes, market mechanism need to consider all relevant constraints to be accepted in practice. In this paper, we argue that airlines face substantial financial constraints. Further, these constraints need to be modeled within the auction mechanism as failing to do so may lead to inefficiency and instability.

In standard auction formats, potential buyers are allowed to place bids for the items they want to obtain. An important feature of any reasonable auction mechanism is to preserve individual rationality of all participants which translates into no buyer having to pay more than their posted bid for the items they receive. While the prices cannot exceed the bids of buyers, they can however be equal to them, e.g. when two buyers place the same bid for an unique item. The celebrated VCG mechanism offers a dominant strategy for all participants to simply declare their true valuations over all items and then payments are calculated in such a way that the prices for each buyer resemble the minimum amount of money required to outbid other bidders (Vickrey 1961). Moreover, since the mechanism has access to the true valuations of all participants, it is possible to derive an efficient allocation with maximal social welfare. However, this result relies heavily on the theories that 1) financial markets are efficient such that buyers can raise enough funding to pay a price up to their valuation and 2) buyers need to have certainty over their true valuation.

In the airline industry, both of these points are unrealistic to assume in practice. Ever since September 11, 2001, airline carriers face severe financial difficulties. In the US, 11 major air carriers merged into four between 2004 and 2015. In Europe, 2017 saw two large airlines go out of business (Air Berlin and Monarch) and Alitalia, the Italian flag carrier, saved from bankruptcy (again) and bought up by the Italian government. As a result of their financial struggles, most airlines face low investor ratings[1], making it more expensive for them to obtain loans. In addition, predicting the cost and value for operating certain flights (and hence requiring specific slots at airports) is notoriously hard. Another financial crisis can make air travel less affordable for the general public, spikes in oil prices can drive up costs, and political climate can make once popular travel destinations unattractive (as has happened to Turkey and Egypt in recent past).

In the face of these difficulties, airlines are either financially constraint or would face significant risk by bidding up to their true (expected) net present value. In most markets, bidders can only submit their budget-capped valuations, which can lead to significant inefficiencies as we show in our computational study. Hence, in order to determine efficient and stable allocation, bidders have to be allowed to submit both, their valuation of packages as well as their financial constraints. Indeed, some markets elicit both, valuations and budgets that must not be exceeded (Nisan et al. 2009), in order to allow bidders to adequately express their preferences and constraints.

The importance of budget constraints in practice has led to significant research in mechanism design. Unfortunately, it was shown that we cannot hope for incentive-compatible mechanisms in the presence of budget constraints in multi-object auctions with private budget constraints (Dobzinski et al. 2008). Incentive-compatibility might be too much to ask for and less of a concern in large markets such as the ones introduced above where participants often have very little information about the preferences of others and strategic manipulation is challenging to say the least. However, even if strategic manipulation is less of a concern, a designer might be interested in stable outcomes, where no participant would want to deviate. Bichler and Waldherr (2019) showed that finding the stable outcome that maximizes welfare requires solving a Σ_2^p-hard optimization problem and described how bilevel programming can be used to derive solutions.

[1] As of the time of this writing, Moody's has rated American Airlines, the worlds largest airline, as Ba3 (speculative, non-investment grade); Lufthansa and British airways, two of the largest European carriers, as Baaa3 (lower medium investment grade).

3 Mathematical Model

Bichler and Waldherr (2019) showed that the allocation and pricing problem for financially constrained buyers is Σ_2^p-hard and proposed an approach based on mixed integer bilevel linear programming (MIBLP). We introduce the general form of a MIBLP, briefly describe the model suggested in Bichler and Waldherr (2019) and explain how we modified the model to reflect the specific conditions in secondary markets for airport slot allocation. We also describe how the model can be adjusted in order to also be used as a mechanism for primary slot allocation.

3.1 Bilevel Programming

Bilevel linear programs are frequently used to model sequential distributed decision making. In these situations, typically a *leader* makes the first decision and a *follower* reacts after observing the leader's decision. The follower's action is important to the leader as it might interfere with the leader's objective. The challenge of the leader is to predict the follower's reaction and take action in such a way that after the follower's reaction the leader's objective is reached to the highest possible degree. More technically, a bilevel linear program is a linear program that is constrained by another linear optimization problem. Usually the first optimization problem is called the upper level problem (leader) while the constraining problem is referred to as the lower level problem (follower). Given an upper level solution, the lower level computes an optimal solution under consideration of its respective constraints. This in turn affects the upper level by altering the value of the objective function or violating constraints, possibly making the overall solution infeasible. Let X be the set of variables in the upper level problem and Y be the set of variables in the lower level problem. The general form of this problem is

$$\max_{x \in X} F(x,y) \tag{1a}$$

$$\text{s.t. } G(x,y) \leq 0 \tag{1b}$$

$$\min_{y \in Y} f(x,y) \tag{1c}$$

$$\text{s.t. } g(x,y) \leq 0 \tag{1d}$$

where $F, f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^1, G : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p, g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^q$ are continuous, twice differentiable functions. Note that in MIBLP, F and f are represented by linear objective functions of the upper and lower level, while G and g are the respective linear constraints. X, Y include continuous as well as integer variables.

A bilevel program can be applied to obtain stable outcomes in combinatorial exchanges of goods. In this case, x is the vector of allocation and payments that are derived by the mechanism in the upper level (the outcome of the upper level). $F(x,y)$ is the social welfare function that needs to be maximized (e.g. gains from trade) and $G(x,y)$ contains all allocation and pricing constraints such that the resulting outcome determined in the upper level is feasible. In the lower level, in return, maximizes the minimal improvement d that is achievable by a member of a blocking coalition, if all of its members would deviate from the outcome determined in the upper level and only trade among themselves.

If a coalition exists such that they could improve upon the upper level outcome, the upper level outcome is instable. Hence, the upper level contains the constraint '$d \leq 0$' which turns the upper level infeasible for all outcomes for which such a blocking coalition exists. Blocking coalitions are determined in the lower level by setting the objective functions to '$\max d$', with the constraint set g determining such a coalition and assuring a feasible outcome y between their members. If d cannot be maximized beyond 0 in the lower level, the upper level outcome is stable since there exists no coalition that can deviate from its outcome such that all members in the coalition can make a profit by doing so.

3.2 Bilevel Programming for Slot Trading

In the following, we describe a bilevel program to model secondary slot trading such that the outcome respects budget constraints and maximizes gains from trade among all stable outcomes. Let I be the set of bidders (airlines) and K be the set of slots available in the exchange. For each bundle of items $S \subseteq K$, $v_i(S)$ defines the true value of bidder i for bundle S. For all $i \in I, S \subseteq K$, let $x_i(S) \in \{0,1\}$ define a binary variable whether i is assigned S in the upper level and let $y_i(S) \in \{0,1\}$ define whether i sells package S in the upper level. For each bidder i, $y_i(S)$ are only defined for packages S that are owned by i. Let $p_i \in \mathbb{R}$ be the corresponding payment for i in the upper level. In case p_i is positive, i receives this payment in the exchange, while she pays $|p_i|$ in case p_i is negative. This is where the model varies substantially from the one presented in Bichler and Waldherr (2019) where the set of buyers and sellers are disjoint (i.e. bidders could not buy and sell at the same time) and payments were strictly non-negative. Below, we show that our bilevel program can easily be adapted to the case with disjoint buyers and sellers (as in the primary allocation of slots) and hence generalizes their model.

Let $d \in \mathbb{R}$ be the minimal profit of a member of a blocking coalition that is determined in the lower level. Further, let $\chi_i(S) \in \{0,1\}$ describe whether i is assigned S in the lower level, $\gamma_i(S) \in \{0,1\}$ describe whether i sells S in the lower level and $\rho_i \in \mathbb{R}$ describe the payment of i in the lower level. For each bidder we also introduce an auxiliary variables a_i to keep track of whether i is a member of the coalition blocking the upper level outcome. Then, the MIBLP looks as follows.

$$\max_{x,y} \sum_{S \subseteq K} \sum_{i \in I} v_i(S) \cdot (x_i(S) - y_i(S)) \tag{CEx}$$

$$\text{s.t.} \sum_{i \in I} p_i = 0 \tag{2a}$$

$$\sum_{S:k \subseteq K} \sum_{i \in I} x_i(S) \leq \sum_{S \subseteq K} y_i(S) \qquad \forall k \in K \tag{2b}$$

$$p_i \geq \sum_{S \subseteq K} \sum_{i \in I} v_i(S) \cdot (y_i(S) - x_i(S)) \qquad \forall i \in I \tag{2c}$$

$$-p_i \leq B_i \qquad \forall i \in I \tag{2d}$$

$$d \leq 0 \tag{2e}$$

$$d = \max d \tag{Lower Level}$$

$$\text{s.t.} \quad \sum_{i \in I} \sum_{S \subseteq K} \rho_i(S) = 0 \tag{2f}$$

$$\sum_{S:k \subseteq K} \sum_{i \in I} \chi_i(S) \leq \sum_{S \subseteq K} \gamma_i(S) \qquad \forall k \in K \tag{2g}$$

$$\rho_i \geq \sum_{S \subseteq K} \sum_{i \in I} v_i(S) \cdot (\gamma_i(S) - \chi_i(S)) \qquad \forall i \in I \qquad (2h)$$

$$-\rho_i(S) \leq B_i \qquad \forall i \in I \qquad (2i)$$

$$a_i \geq \chi_i(S) \qquad \forall S \subseteq K, i \in I \qquad (2j)$$

$$a_i \geq \gamma_i(S) \qquad \forall S \subseteq K, i \in I \qquad (2k)$$

$$d \leq M \cdot (1 - a_i) + \sum_{S \subseteq K} v_i(S) \, (\chi_i(S) - \gamma_i(S)) + \rho_i - \left[\sum_{S \subseteq K} v_i(S) \, (x_i(S) - y_i(S)) + p_i \right] \qquad \forall i \in I \qquad (2l)$$

$$\sum_{i \in I} a_i \geq 1 \qquad (2m)$$

$$\chi_i(S), \gamma_i(S) \in \{0,1\} \qquad \forall S \subseteq K, i \in I \qquad (2n)$$

$$a_i \in \{0,1\} \qquad \forall i \in I \qquad (2o)$$

$$\rho_i(S) \in \mathbb{R} \qquad \forall S \subseteq K, i \in I \qquad (2p)$$

$$d \in \mathbb{R} \qquad (2q)$$

$$x_i(S), y_i(S) \in \{0,1\} \qquad \forall S \subseteq K, i \in I \qquad (2r)$$

$$p_i(S) \in \mathbb{R} \qquad \forall S \subseteq K, i \in I \qquad (2s)$$

(**CEx**) determines an outcome that maximizes the gains from trade under all feasible and stable outcomes. The upper level constraints include economic pricing and allocation constraints such as budget balance (2a), balance of supply and demand (2b), individual rationality (2c), and respects the budget constraints of all bidders (2d). Constraint (2e) states that the solution is infeasible (unstable) if the lower level finds a coalition with positive deviation value d.

Given the upper level solution, the lower level problem tries to find a coalition that can deviate in a beneficial way from the upper level. This is modeled as a maximization problem similar to the upper level. Trades in the lower level also have to fulfill all economic pricing and allocation constraints (2f)-(2i).

Constraints (2j) and (2k) set $a_i = 1$ if bidder i performs a trade in the lower level. In this case, i is member of a blocking coalition. Constraints (2l) models the gains from deviation for each individual bidder, comparing her outcomes (and hence her payoff) in the upper and the lower level. Only if $i \in I$ actually participates in a blocking coalition (i.e. when $a_i = 1$), her difference in payoffs should be considered. We define a very large number M such that d is not affected by bidders that are not members of the blocking coalition. To avoid empty coalitions with only trivial bounds on the d value an constraint (2m) ensures that at least on participant makes a trade in the lower level. The corresponding domains of the upper and lower level variables are defined in (2n) - (2s).

We briefly show how the MIBLP from the preceding section can easily be adapted for primary slot allocation. In the primary market, each bidder is either a buyer (airline) or a seller (airport). Hence, we can simply ignore all variables x_i and χ_i for sellers i as well as all variables y_i and γ_i for buyers i. The remaining constraints and variables can be used in the same way as in (**CEx**). Additional constraints (grandfather rights, etc.) can be easily incorporated as well.

4 Algorithmic Solution

Bard and Moore (1990) initiated algorithmic solutions to MIBLPs. Their algorithm converges if either all leader variables are integer, or when the follower subproblem is an LP. Until recently, MIBLPs were considered "still unsolved by the operations research community" (Delgadillo et al. 2010). Only two years ago, two general purpose branch-and-cut MIBLP algorithms have been proposed by Fischetti et al. (2017) and Tahernejad et al. (2017). Fischetti et al. (2017) extend their earlier algorithm for MIBLPs with binary first-level variables to problems where linking variables are discrete. Tahernejad et al. (2017) propose another general-purpose MIBLP solver based on branch-and-cut which is available open source in the MibS solver. The latter requires the linking variables, those variables that have non-zero coefficients and are present in the upper- and lower-level program, to be integer.

If the lower level problem does not contain integer variables and is an LP, bilevel programs can be reformulated as single-level problem by replacing the lower level with its optimality conditions (e.g., Karush-Kuhn-Tucker (KKT)) and then solving the resulting problem (Bard and Moore 1990). For general mixed-integer bilevel problems, in which linking variables can be both, integer and continuous, Zeng and An (2014) proposed a column-and-constraint generation framework that fixes integer variables in the lower level (thereby transforming it into an LP) and then adds optimality conditions for these solutions to the upper level problem. Bichler and Waldherr (2019) extend this framework for problems in which the lower level is always feasible regardless of the upper level and the objective of the lower level is to make the upper level (and hence the whole MIBLP) infeasible. This is the case in (**CEx**) since simply making the same trades as in the upper level is always feasible for the lower level and the upper level is feasible if and only if an objective value of $d > 0$ is obtained in the lower level.

In the following, we give a short outline of the algorithm. First, the upper level U of the bilevel program is solved, ignoring all lower level variables and constraints, resulting in optimal allocations x^*, y^* and prices p^*. Afterwards, the lower level is solved for the optimal upper level outcome. If for the optimal solution d^* of the lower level, it holds that $d^* < 0$, the upper level outcome is stable. Otherwise, a blocking coalition can be read from the lower level variables χ^*, γ^*, ρ^*. In this case, the KKT conditions for the fixed variables χ^* and γ^* are added to the upper level U and the upper level is solved again. The KKT conditions assure that in this iteration U will return an outcome that is stable against a coalition with assignments χ^*, γ^*. The process is repeated iteratively until either no more blocking coalition can be determined in the lower level (in which case the last upper level outcome is the one that maximizes welfare among all stable outcomes) or U is no longer feasible itself (in which case there is no stable outcome).

Iteratively adding KKT conditions to the upper level adds immensely to an already large integer program as in each iteration we have to add a large number of lower level primal and dual constraints and variables, as well as the complementary slackness conditions. The latter, especially, complicate the program since they either require quadratic constraints or linearization by using even more big M constraints. In the following, we describe an alternative to adding KKT conditions that vastly reduces the number of necessary constraints and variables while ensuring stability against the same blocking coalitions in each iteration. In comparison to KKT conditions, we refer to this set of constrains as the blocking coalition elimination (BCE) constraints.

The main idea behind these constraints is that we can calculate the amount a buyer is willing to pay in a blocking coalition C and what payment sellers demand in order to deviate. For both buyers and sellers we can calculate their utility π in the upper level solution. For a buyer this is her valuation for the allocated bundle minus the price she has to pay. Similarly, a seller's utility is the payment she receives minus her valuation for the bundle she sold. If a participant acts in both roles, her utility is the sum of the aforementioned terms (UB). To indicate the change in utility a participant experiences when deviating with the given coalition let the constant v_i^C denote the utility a participant has when buying and selling items in C. Then, in a coalition with given allocation, a participant is willing to pay the difference of v_i^C (her utility given the lower level allocation) and π (her utility given the current upper level allocation and prices) in combination with a small ε she wants to gain by deviating (WtP). Note that her willingness to pay w_i^C is still constrained by her budget (WtP). While this min function also results in either quadratic constraints or the necessity of linearization, the overhead is way lower than for KKT conditions. If a participant by the allocation alone has negative utility $v_i^C < 0$ in the coalition she will instead demand a payment of amount w_i^C instead. Hence, if the sum of w_i^C is negative participants in the coalition demand more money for selling their items than the other participants are willing to pay for these items and the coalition will not deviate (Ex).

$$\pi_i = \sum_{S \subseteq K} v_i(S)\,(x_i(S) - y_i(S)) + p_i \qquad \forall i \in C \qquad \text{(UB)}$$

$$w_i^C = \min\{B_i, v_i^C - \pi_i - \varepsilon\} \qquad \forall i \in C \qquad \text{(WtP)}$$

$$\sum_{i \in C} w_i^C \leq 0 \qquad \text{(Ex)}$$

$$w_i^C \in \mathbb{R} \qquad \forall i \in C \qquad \text{(Real)}$$

The careful reader might wonder whether a coalition would deviate when a participant that only has the role of a buyer demands a payment in the coalition (because the upper level allocation and prices are preferable to her). One could argue that this coalition would not form and that since we are not accounting for this case we restrict the upper level in inadmissible ways. However, if (Ex) is still violated, i.e. the coalition collectively is willing to pay enough money to cover for the demanding buyer, a smaller coalition could form by dropping the buyer in question. The constraints in this sense not only ensure stability against coalitions with known allocations but also some similar coalitions whose formation is predictable. After these constraints have been added to the upper level based on a lower level allocation, all solutions that satisfy these constraints are resistant to deviations with the given or very similar allocations. Similar to the general framework with KKT conditions, iteratively adding BCE constraints either leads to an outcome for which there no longer exists a blocking coalition (a stable outcome) or to infeasibility in the upper level (proving non-existence of a stable outcome).

5 Experimental Results

In this section, we provide evidence that our mechanism is computationally efficient and improves the welfare in secondary slot trading markets as opposed to mechanisms that ignore budget constraints. We describe the data set that we used to evaluate our mechanism in Section 5.1. Afterwards, we analyze the benefits of using the new BCE constraints in comparison to the classical KKT conditions in Section 5.2 before showing that with the help of these new constraints our mechanism is capable of solving instances of practically relevant size in Section 5.3. Finally, we demonstrate the necessity of using our mechanism that elicits budget constraints in combinatorial exchanges in Section 5.4. All algorithms were implemented in Java. The experiments were executed on a laptop with Intel core i5-6600k (4 cores, 3.5 GHz) and 8GB RAM. Gurobi 7.5.2 was used for solving linear programs.

5.1 Data

The set of instances depicts a secondary market where all bidders are buyers and sellers at the same time. To model the bids of buyers we used the approach suggested in the combinatorial auction test suite (CATS) (Leyton-Brown et al. 2000). In this, participants are interested in buying a random flight between 2 of the 4 coordinated US airports. Additionally, participants place bids on flights where takeoff/landing times deviate by a small margin from their preferred flight. This flights are adjacent in a temporal sense but travel between the same origin-destination pair as the preferred flight. The valuation of the preferred flight is based on common valuations of the slots and a private random deviation. Valuations for similar flights are based on the valuation of the preferred flight and the degree of deviation. Unfortunately, buyers' budgets are not considered in CATS. We therefore cautiously extended the approach by CATS by randomly generating a buyer i's budget in the interval of half her highest valuation v_i^{max} and her highest valuation $[\frac{1}{2}v_i^{max}, v_i^{max}]$. On the seller side, CATS does only consider a single seller without ask prices. However, secondary markets feature several sellers that could potentially use a slot that is not sold in the exchange for themselves and thus have ask prices for the slots they offer. We therefore implemented a general mechanism that assigns slots to sellers and computes ask prices based on the valuations proposed by CATS. We distribute slots in such a way that the number of items per participant is approximately uniform, but choose the assigned slots randomly. According to the buyer case, valuations for these items were generated by common valuations and private value deviation. To model the participants willingness to sell these items, we further applied a factor of 0.3 to compute the final ask prices. The exact number of instances for a specific parameter configuration will be discussed in the appropriate sections of this chapter.

5.2 Comparison of KKT to BCE conditions

First, we evaluate the benefits of the BCE introduced in Section 4 as opposed to using the standard KKT conditions. For this, we consider two settings with 10 and 15 bidders, respectively. In both scenarios, 40 slots are traded. We solved 10 instances each, one time using standard KKT conditions, the other time using our BCE constraints. The results are depicted in Table 1. Out of 10 instances with 10 bidders that both buy and sell and 40 items we can solve 7 when using KKT conditions and all 10 when using BCE constraints. Even if we consider only the times of solved instances for the averages, the KKT conditions lead to significantly larger runtimes (23.57s compared to 1.87s). The differences become even more evident when we consider instances with 15 bidders and 40 items. Using KKT conditions not a single instance could be solved within 300s while with BCE all instance in 11.7s on average could be solved.

#Bidders	#Slots	Method	avg. runtime of solved instances (s)	solved <300s	avg. #iterations
10	40	KKT	23.57	7	11.3
10	40	BCE	1.87	10	14.2
15	40	KKT	n.a.	0	n.a.
15	40	BCE	11.70	10	33.2

Table 1: Comparison of KKT and BCE constraints

The examples showed that BCE constraints perform significantly better than the KKT conditions for the scenarios under consideration. We attribute these gains in time to the lower number of constraints in general and the lower number of big M constraints, specifically.

5.3 Computability

In the following, we show that our approach can solve instances of practically relevant size despite the high complexity of the underlying problem. Let us first discuss how many participants realistically compete against other in secondary slot trading markets. The world's largest airline alliance, Star Alliance, consists of 27 members[2]. Since only a handful of such alliances exists, almost all airlines are member in one of them and members of one alliance only compete to a certain degree in the same region, we argue that usually not more than 25 airlines will compete in the same market. The ACL reported that the annual trade volume at Heathrow was around 250 slots from 2014 to 2016[3]. Since slots are usually traded for every day of the week, this number is essentially inflated by a factor of 7. This leaves us with around 30 annually traded slots at one of the busiest airports in the world. For this computational study we assume that between 100 and 200 total slots from 4 different airport are traded. This scenario matches the number of coordinated airports in the US and the number of slots offered per airport is around $83 - 166\%$ of slots traded at Heathrow in recent years.

For these experiments we generated instances each for 20, 25 and 30 bidders. We incremented the number of slots in steps of 20 from 100 to 200 (inclusive) and generated 2 instances per number of slots leading to a total number of 12 instances per participant count. The number of slots offered did not lead to significant differences in computational time, therefore we report the aggregated results for the 12 instances each in Table 2. For 20 to 30 bidders, we report how many of the 12 instances we were able to solve (i.e. obtain a welfare-maximal stable outcome or proof that no such outcome exists) within a time limit of 1 hour. We also report the average computational time (in seconds) of the solved instances. It can be seen that all of the smaller instances could be solved in very short time and our approach was still able to obtain solutions for 7 out of the 12 largest instances, providing welfare-maximal outcomes that are stable against blocking coalitions of all sizes. In practice, however, blocking coalitions of arbitrary large size may not form since coordinating towards blocking an outcome may be hard for large groups of (self-interested) airlines. Hence, we also determined welfare-optimal outcomes that are only stable against blocking coalitions of size at most 3 (3-stable) or 5 (5-stable). It can be seen, that our approach is well capable of obtaining

[2]https://www.staralliance.com
[3]https://www.heathrow.com/file_source/ HeathrowNoise/Static/HCNF_ACL_Slot_Coordination.pdf

outcomes against coalitions of these sizes, solving all of the larger instances in only a matter of seconds. While solving instances of increasing size becomes computationally more challenging, these results indicate that 3- and 5-stable outcomes can be obtained for an even larger number of bidders, should the policy maker be satisfied by this level of stability.

#Bidders	Stable Outcome		3-Stable Outcome		5-Stable Outcome	
	#Solved	Runtime (s)	#Solved	Runtime (s)	#Solved	Runtime(s)
20	12	67	12	1	12	2
25	9	382	12	1	12	3
30	7	2426	12	1	12	8

Table 2: Computability of secondary market instances

5.4 Inefficiencies with Capped Bid

Incorporating budget constraints into the allocation and pricing mechanism comes at the cost of a more complex algorithm. Finding core prices for allocation problems without budget constraints does not require solving a Σ_2^p-complete optimization problem, but 'only' a NP-complete problem (Lehmann et al. 2006). However, we show in the following that ignoring budget constraints may lead to really bad outcomes of the mechanism.

When bidders cannot communicate their financial constraints, it is risky for them to state their true valuations if these are above their budgets. In order to avoid prices that they cannot pay, bidders' reported values and bids need to therefore be capped at their budget. This also makes it hard for bidders to communicate differences in valuations for bundles that are both above their budget. In case they report two capped bids, the corresponding bundles appear to have the same associated valuation while in reality the true valuations might differ substantially. In the following, we refer to the scenario where bidders only submit their capped bids as the capped market, and to the scenario where they communicate their true valuations and budgets as the uncapped market.

In order to investigate the welfare loss incurred by inability to communicate budgets, we analyzed markets with 10, 15 and 20 bidders as well as 40 and 50 slots traded. For each combination of bidders and slots, we solved 10 instances, determining a welfare-maximal stable outcome for the capped as well as for the uncapped market. We calculate the loss in welfare by comparing the sum of true (uncapped) payoffs achieved by the winners in the capped market to the sum of payoffs achieved by winners in the uncapped market. Table 3 summarizes the results, showing for each combination of bidders and traded slots the percentage of the uncapped market's welfare that is achieved in the capped market. We report the average welfare over the ten instances as well as the worst and best welfare achieved in the capped market for these ten instances. The average welfare loss in secondary markets ranged from 15% to 20% (Table 3). For no instance can the capped market find an outcome that achieves 100 % of the welfare of the uncapped market. In some instances, the welfare loss is even higher than 50%.

In addition to these severe losses in welfare, not being able to communicate budgets can also lead to instabilities. An outcome that is stable in the capped market may not be stable when the true valuations of bidders are considered. To analyze this effect, we determined a welfare-maximal stable outcome in the capped market and then tested whether a blocking coalition forms when

#Bidders	#Slots	Avg Welfare	Worst Instance	Best Instance
10	40	0.84	0.61	0.93
10	50	0.79	0.46	0.97
15	40	0.80	0.51	0.91
15	50	0.84	0.68	0.98
20	40	0.84	0.56	0.97
20	50	0.85	0.74	0.89

Table 3: Welfare losses due to capped bidding

true valuations are considered. Table 4 summarizes the results for the same instances we used for the analysis of the welfare losses. Almost all computed solutions based on capped valuations are unstable when considering real valuations, even for small instances with 10 bidders. We report for how many of the instances, a blocking coalition would form against the outcome of the capped market given the true valuations. It can be seen that over all 60 instances, only a single outcome is also stable in the uncapped market. Even worse, we also report how often reasonable small coalitions (of size up to 5 or 3) block the capped market's outcome. While it may be unrealistic for larger coalitions to form, these small coalitions constitute a significant risk to stability. Determining allocations and prices based on capped valuations thus seem impractical in general as bidders almost always have an incentive to deviate from the outcome.

#Bidders	#Slots	Instances with blocking coalitions		
		in general	of size up to 5	of size up to 3
10	40	9	7	3
10	50	10	10	6
15	40	10	10	5
15	50	10	8	6
20	40	10	10	9
20	50	10	10	7

Table 4: Instability due to capped bidding

6 Conclusion

Airport time slots are the legislative answer to growing air traffic and congested airports. While markets for secondary trading of time slots are implemented in practice, they fail to take budget constraints of the bidders into account. We argue that budget constraints play an important role in practical applications and propose a model for budget constrained secondary markets based on a MIBLP. We conducted experiments on instances of realistic size to show that contrary to popular belief some MIBLPs are tractable in practice. In computational experiments we also proved that the ability to communicate budget constraints is of crucial importance. Our experiments show that ignoring budgets leads to substantial loss in social welfare, up to 50% in extreme cases. Additionally, outcomes determined based on capped bids also lead to severe instabilities and market bidders would form blocking coalitions due to their true valuations. We conclude that budget constraints should be incorporated into a mechanism for airport time slot allocation to find allocations and prices that are feasible in practice. To this end, we were able to show that our approach based on bilevel programming can be used as an efficient mechanism.

References

Adomavicius, G., S. Curley, A. Gupta, P. Sanyal. 2013. Impact of information feedback in continuous combinatorial auctions: An experimental study of economic performance. *MIS Quarterly* **37** 55–76.

Adomavicius, G., A. Gupta. 2005. Toward comprehensive real-time bidder support in iterative combinatorial auctions. *Information Systems Research (ISR)* **16** 169–185.

Avenali, Alessandro, Tiziana D'Alfonso, Claudio Leporelli, Giorgio Matteucci, Alberto Nastasi, Pierfrancesco Reverberi. 2015. An incentive pricing mechanism for efficient airport slot allocation in europe. *Journal of Air Transport Management* **42** 27–36.

Ball, Michael, George Donohue, Karla Hoffman. 2006. Auctions for the safe, efficient, and equitable allocation of airspace system resources. *Combinatorial auctions* **1**.

Ball, Michael O, Frank Berardino, Mark Hansen. 2017. The use of auctions for allocating airport access rights. *Transportation Research Part A: Policy and Practice* .

Bard, Jonathan F, James T Moore. 1990. A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing* **11**(2) 281–292.

Bichler, M., A. Gupta, W. Ketter. 2010. Designing smart markets. *Information Systems Research* **21**(4) 688–699.

Bichler, Martin, J Goeree. 2017. *Handbook of Spectrum Auction Design*. Cambridge University Press.

Bichler, Martin, Stefan Waldherr. 2019. Computing core-stable outcomes in combinatorial exchanges with financially constrained bidders. *Proceedings of the 2019 ACM Conference on Economics and Computation*. EC '19, ACM, New York, NY, USA, 747–747.

Bikhchandani, Sushil, Joseph M Ostroy. 2002. The package assignment model. *Journal of Economic Theory* **107**(2) 377–406.

Castelli, Lorenzo, Paola Pellegrini, Raffaele Pesenti. 2011. Airport slot allocation in europe: economic efficiency and fairness. *International journal of revenue management* **6**(1-2) 28–44.

Delgadillo, Andrés, José Manuel Arroyo, Natalia Alguacil. 2010. Analysis of electric grid interdiction with line switching. *IEEE Transactions on Power Systems* **25**(2) 633–641.

Dobzinski, Shahar, Ron Lavi, Noam Nisan. 2008. Multi-unit auctions with budget limits. *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*. IEEE, 260–269.

Fan, Ming, Jan Stallaert, Andrew B Whinston. 2003. Decentralized mechanism design for supply chain organizations using an auction market. *Information Systems Research* **14**(1) 1–22.

Fischetti, Matteo, Ivana Ljubić, Michele Monaci, Markus Sinnl. 2017. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research* .

Goetzendorff, A., M. Bichler, P. Shabalin, R. W. Day. 2015. Compact bid languages and core pricing in large multi-item auctions. *Management Science* **61**(7) 1684–1703.

Guo, Zhiling, Gary J Koehler, Andrew B Whinston. 2012. A computational analysis of bundle trading markets design for distributed resource allocation. *Information Systems Research* **23**(3-part-1) 823–843.

Jones, I, S Holder, JP VAN DER VEER, E Bulman, N Chesters, S Maunder, P Kalmus, S Sorensen, F Maiorano, PM de Leon, et al. 2004. Study to assess the effects of different slot allocation schemes. Tech. rep., NERA Economic Consulting Reort for the European Commission.

Lehmann, Daniel, Rudolf Müller, Tuomas Sandholm. 2006. The winner determination problem. *Combinatorial auctions* 297–318.

Leyton-Brown, Kevin, Mark Pearson, Yoav Shoham. 2000. Towards a universal test suite for combinatorial auction algorithms. *Proceedings of the 2nd ACM conference on Electronic commerce*. ACM, 66–76.

Ltd., Dot Econ. 2001. Auctioning airport slots: A report for hm treasury and the department of the environment, transport, and the regions .

Madas, Michael A., Konstantinos G. Zografos. 2010. Airport slot allocation: a time for change? *Transport Policy* **17**(4) 274 – 285.

Martin, Alexander, Johannes C Müller, Sebastian Pokutta. 2014. Strict linear prices in non-convex european day-ahead electricity markets. *Optimization Methods and Software* **29**(1) 189–221.

Nisan, Noam, Jason Bayer, Deepak Chandra, Tal Franji, Robert Gardner, Yossi Matias, Neil Rhodes, Misha Seltzer, Danny Tom, Hal Varian, et al. 2009. Google's auction for TV ads. *International Colloquium on Automata, Languages, and Programming*. Springer, 309–327.

Pellegrini, Paola, Lorenzo Castelli, Raffaele Pesenti. 2012. Secondary trading of airport slots as a combinatorial exchange. *Transportation Research Part E: Logistics and Transportation Review* **48**(5) 1009–1022.

Rassenti, Stephen J, Vernon L Smith, Robert L Bulfin. 1982. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics* 402–417.

Roth, Alvin E. 2002. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica* **70**(4) 1341–1378.

Tahernejad, Sahar, Ted K Ralphs, Scott T DeNegre. 2017. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Working Paper* .

Vickrey, William. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* **16**(1) 8–37.

Walsh, W. E., M. P. Wellman, F. Ygge. 2000. Combinatorial auctions for supply chain formation. *Proceedings of the 2Nd ACM Conference on Electronic Commerce*. EC '00, ACM, New York, NY, USA, 260–269. doi:10.1145/352871.352900. URL `http://doi.acm.org/10.1145/352871.352900`.

Zeng, Bo, Yu An. 2014. Solving bilevel mixed integer program by reformulations and decomposition. *Optimization online* 1–34.

5 Online Pricing

Peer-Reviewed Conference Paper

Title: Real-Time Dynamic Congestion Pricing: An Online Optimization Approach

Authors: R. Littmann, M. Bichler

In: Workshop on Information Technologies and Systems 2020

Abstract: Traffic congestion is the central problem for the future of urban mobility. While a growing number of cities introduces static congestion pricing, these pricing schemes are widely considered inefficient. They are too low at some times to avoid congestion and too high at other times leading to low adoption and a negative impact on welfare. Now, mobile devices are widely available, and the technology is ready to implement real-time congestion pricing. However, we do not have adequate models and algorithms to price road capacity in real-time yet. Recent advances in optimization can provide the foundation for new information systems to price trips in real-time. We leverage new developments in online optimization and introduce an algorithm to update potentially thousands of prices for roads in a city in real-time. We analyze the approach with the widely used SimMobility traffic simulator and show a substantial decrease in traffic jams and an increase in welfare.

Contribution of thesis author: Results, implementation, joint project and paper management

Reference: Bichler and Littmann (2020)

Real-Time Dynamic Congestion Pricing: An Online Optimization Approach

Richard Littmann[1,2] Martin Bichler[1]

[1] Technical University of Munich, Munich, Germany
[2] Advanced Optimization in a Networked Economy (AdONE), GRK 2201, Technical University of Munich, Leopoldstr. 145, 80804 Munich, Germany

{richard.littmann,bichler}@in.tum.de

1 Introduction

Mobility-as-a-Service (MaaS) and transportation sharing platforms are expected to be the future of transportation, at least in urban areas. The prices for transportation are decreasing even further with such new technologies and increasing competition for mobility and transportation services. Overall, the global on-demand transportation market is expected to reach \$290.3 billion by 2025.[1] Together with increased urbanization, this will lead to more and more traffic in major cities, and, as a result, more and more congestion: one of the major problems of our society. Costs due to traffic jams worldwide were estimated \$1 trillion already in 2013 in an article by the Economist. In the U.S., congestion cost was at €87 billion in 2018.[2] Those numbers come from the lost productivity of workers sitting in traffic, and wasted fuel, among other factors. Not surprisingly, traffic takes the biggest economic toll on the largest, most economically vibrant cities.

A central challenge to traffic jam prevention is mispricing of the core resource for mobility, the road capacity. In most cities, road capacity is not priced at all. A combination of MaaS, car sharing apps, and congestion pricing has the potential to alleviate the problems that have emerged due to traffic jams. Congestion prices are charged to drivers who enter roads in high demand and can lead individuals to carpool, possibly in a self-driving car fleet, to switch to public transit, or cancel a leisure trip if roads are already congested. Even in the absence of MaaS and car sharing apps, this can lead to a much better utilization of scarce road capacity. Congestion pricing therefore also reduces private car travel demand and the associated carbon dioxide emissions (Wang et al. 2014).

A few cities such as London, Milan, Stockholm, or Singapore have introduced road pricing to combat congestion, and several more are about to follow suit soon (e.g. Beijing in 2020, New York in 2022, Sao Paulo in 2025). The pricing systems largely fall into three categories: zonal-based (sometimes called area-based) pricing for being inside an area, cordon-based pricing for passing a cordon line, and distance-based pricing (Cheng et al. 2017). The zonal-based pricing scheme was already established in 1975 in Singapore. The Electronic Road

[1] https://www.economist.com/the-economist-explains/2014/11/03/the-cost-of-traffic-jams
[2] http://inrix.com/scorecard/

Pricing (ERP) system, which was introduced in 1998, works with other information systems to manage traffic congestion and improve transportation efficiency. The traffic volume in the toll area was reduced by 45%, which was considered a huge success. A cordon-based pricing scheme was introduced in central London in 2003 and in Stockholm in 2006. Most of the existing congestion pricing models are based on static prices (Chiu et al. 2011; de Palma and Lindsey 2011).

However, efficient congestion prices need to adapt to changing traffic conditions to price at marginal costs (Cheng et al. 2017). Static models only account for the network flows in a stationary manner, ignoring the time-dependent nature of traffic flows and travel behavior. Lo and Szeto (2004) showed that transportation management schemes based on static traffic assignment theory can actually worsen the congestion problem. Prices might be too high or too low to avoid congestion. In addition, static congestion pricing models do not consider the temporal effect of the current toll on future congestion (Wie and Tobin 1998). Singapore's ERP differentiates by vehicle type, time of day, and location of tolling gantry and comes closest to a dynamic congestion pricing scheme in practice today. Such prices are typically not determined in real-time and do not respond to sudden changes in demand or the length and specific route of a driver. All implemented congestion pricing schemes adopt the flat-toll method regardless of the travel distance, travel time, or congestion level in the pricing zone. Hence, this toll method may cause inequitable charging problems due to undercharging for long journeys and overcharging for short ones. Cheng et al. (2017) write: "How to formulate distance-based congestion toll encapsulating all facets of a dynamic transportation network while alleviating urban traffic congestion remains an open question."

New technology such as mobile apps and privacy-preserving tracking of vehicles allows for real-time dynamic pricing today (Hensher 2018). Compared with conventional static congestion pricing models, real-time congestion pricing models are much more efficient due to the inherent uncertainty, randomness, and time-varying properties of traffic. However, real-time congestion pricing requires new algorithms to best adapt to changing market conditions and new information systems designs. Following the theme of this year's workshop, we provide a multi-method design for an intelligent information system for social good: congestion-free traffic.

1.1 Scenario

We consider a scenario where a service operator or regulator "owns" capacity on roads in a city. The service provider sells her inventory to end users such as daily commuters in private cars, taxis, ride-hailing providers, or professional carriers and craftsmen who need to visit their clients.[3] The objective of the service provider is to price individual roads over time such that the road capacity is not exceeded and no congestion emerges. While the service provider might want to provide long-term plans for regular trips (e.g., the daily commute), a large part of the trips needs to be priced dynamically in real-time, and these prices need to adjust to changing traffic conditions. Without congestion, consumers would drive for free, but with higher demand prices shall rise. In this scenario, a consumer wants to drive from

[3] Protected groups get reduced prices for regular trips they have to make. Examples would be single moms getting their kids to childcare, or retired persons going to recreational events. We do not further consider equity issues in this paper.

a certain origin to a destination. She submits a query to the service provider online, who responds with a number of possible routes and prices for each of these routes.

For occasional trips, drivers can also search ahead of time and lock in a price to drive from an origin to a destination within a certain time frame. If the customer does not start her trip within the time slot scheduled (either immediately or in some future time slot), she will also not be charged or only be charged a minimal fee. If she has to terminate her trip before reaching the destination (e.g., due to an accident), she will only be charged the roads driven.

The key challenge is that the service provider needs to compute a price for hundreds of thousands of trips dynamically. Real-time dynamic pricing requires very fast algorithms because the prices must be updated frequently to adapt to changing demand patterns. Real-time pricing algorithms that consider the time and the route of the trip have not yet been discussed in the literature to our knowledge. If all bids were available at one point in time, the decision maker would need to solve a very large-scale optimization problem. This problem aims to maximize welfare and decides which road trips to accept. If the problem was convex, dual prices could be interpreted as market prices of such an offline optimization problem. However, the problem has to be solved dynamically as new drivers arrive, and it is a non-convex integer optimization problem. We draw on recent advances in online linear programming (Agrawal et al. 2014; Li and Ye 2019; Li et al. 2020) and combine them with short term forecasting in order to provide good estimates for parameters needed in the online integer program. We solve the problem in dual space, and compute dual prices that reflect the scarcity of the resources, as we discuss next.

1.2 Pricing Roads in Real-Time

In our problem, the individual resources can be seen as roads in an urban road network. Each road leads to a capacity constraint of an optimization problem, a maximum number of vehicles that must not be exceeded during a time slot. A route for a driver can be modeled as a 0-1 column vector in an online integer linear program that has strictly positive entries on all the roads (components of the vector) on this route and 0 otherwise. The dual prices of each capacity constraint reflect the opportunity costs of another car on this road in a time slot. The sum of the dual prices of all roads on a route then results in the price of this route.

The question is how to react to demand changes over time and adapt prices in real-time. This requires solving a large integer packing problem dynamically. Very recently, Li et al. (2020) provided a fast optimization algorithm based on projected stochastic subgradient descent to address this task. Stochastic gradient descent is widely used for training neural networks and is known to be very fast. As such, the algorithm has potential for pricing roads in real-time for realistic problem sizes with thousands of roads. The algorithm outputs an integer solution with an expected regret of $O(m\sqrt{n})$ under the stochastic input model, which is a measure in online optimization for how well an algorithm is doing compared to the best dynamic decisions in hindsight. In our context, m would be the number of roads, and n the number of trips. The algorithm can be seen as integer approximation of the offline integer linear program. The online integer program makes sure that resources are not depleted too early and adapts to demand peaks in contrast to a fixed price that a service provider would set during the time slot.

Two parameters are essential in online integer programming: the value or willingness-to-pay v that each bidder has for a trip, and the number n of trips (or requests) we expect to arrive over time. While we do not know the number of vehicles which will drive on a road within this time slot exactly, we can get very precise forecasts for an hour or less. This forecast can be done just before the hour starts in order to be as precise as possible. A big advantage of the algorithm suggested in this paper is that we do not need to elicit the precise willingness-to-pay from a driver. The system generates a price and the driver accepts or rejects this price, which is all the information needed. Therefore, no driver needs to reveal his precise value v, she just needs to accept or reject an offer at a price. This is important as many drivers might want to keep their willingness-to-pay private information. This is similar to ascending auctions following the primal-dual design recipe, where winning bidders do not have to reveal their private values (Bikhchandani et al. 2011; Bichler 2017). Yet, without knowing all valuations, primal dual auctions maximize welfare. The property also preserves privacy of the drivers.

1.3 Contribution

Computing congestion prices in real-time that take into account the time and the very route of a driver has been an open problem. Solving such a problem will enable efficient pricing of road capacity and ultimately mitigate or eliminate traffic congestion. We introduce a dynamic algorithm based on short-term forecasting and stochastic gradient descent which allows us to update tens of thousands of prices in real-time based on changing demand. We provide an experimental evaluation of dynamic congestion pricing based on the SimMobility framework[4]. This software framework is a state-of-the-art tool used by the traffic engineering community to benchmark and analyze traffic in a mid-sized artificial city (Adnan et al. 2016). The results show that the algorithm scales to the problem sizes required in practice, effectively reduces traffic jams, and increases welfare compared to static pricing or no pricing.

1.4 Outline

The paper is organized as follows: In Section 2, we review prior literature on congestion pricing. We describe the model under consideration in Section 3 and the online algorithm in Section 4. In Section 5, we provide a brief introduction to the SimMobility framework. An experimental evaluation is given in Section 6. Finally, we conclude in Section 7.

2 Related literature

Maximizing social welfare requires efficient pricing, i.e. the cost of a trip must equal its marginal social cost. Static pricing schemes are inefficient because they generally are not demand-responsive. Drivers may either pay much more than necessary or too little to avoid congestion. Prior literature agrees that dynamic congestion pricing schemes offer the most efficient approach to manage and operate roads (de Palma and Picard 2005; Rouhani and Niemeier 2014). Dynamic congestion pricing (DCP) is the determination of time-varying

[4] `https://github.com/smart-fm/simmobility-prod`

prices to control or otherwise reduce traffic congestion, and hence improve social welfare (Do Chung et al. 2012; Cheng et al. 2017). Nobel laureate William Vickrey was the first to suggest congestion prices and he already suggested dynamic and real-time prices (Vickrey 1963). He argues that "the results of not charging motorists for their rush-hour usage can be disastrously expensive."

The existing literature on DCP can be categorized in within-day and day-to-day models which aim to determine time-varying tolls. Many of these models are formulated as bi-level programming models or mathematical programming models with equilibrium constraints with the objective of minimizing total system travel times and the constraints of dynamic route choice and departure time decisions of each individual.

DCP is not necessarily real-time and does not adapt to changes in demand. New technology including mobile applications and GPS-based and privacy-preserving tracking of vehicles allow for real-time dynamic pricing today (Hensher 2018). The analysis of real-time congestion prices has been focused on simple networks (Laval et al. 2015). In contrast, we aim to compute real-time dynamic prices for the entire road network of a city taking into account the length and very route of a trip. For this, we draw on recent advances in online linear programming (Li and Ye 2019). These algorithms have proven regret bounds and allow for computation and real-time adaptation of prices in very large-scale road networks.

3 The Model

Let $N(V,E,c)$ be a directed graph where $c : E \to \mathbb{R}_0^+$ denotes the capacity of an edge or road segment $e \in E$, i.e. the vehicles that can drive on this road segment per time slot without congestion. Drivers want to take trips between an origin-destination pair and bid routes $S \subseteq E$ with $m = |E|$. There are n drivers $i \in \mathscr{I}$. For each route S a driver i has a willingness-to-pay $v_i(S) \geq 0$. We use parameters a_i to denote whether a road segment e is part of a route S for driver i. These parameters might also be used to describe different types of vehicles (e.g., a truck or car). For our basic model, we treat all vehicles equal. The vector $a_i \in \{0,1\}^m$ describes all the roads visited on a route by a driver i. By decision variable $x_i(S) \in \{0,1\}$ we denote whether driver i is assigned the route S and by $p(e)$ the price for road $e \in E$. The integer linear programming formulation for the offline version of the resulting resource allocation problem is straightforward.

$$\max \sum_{S\subseteq E} \sum_{i\in\mathscr{I}} v_i(S)x_i(S) \qquad \text{(Primal)}$$

subject to

$$\sum_{S\subseteq E} \sum_{i\in\mathscr{I}} a_i x_i(S) \leq c(e) \qquad \forall e \in E \ \ (p(e)) \qquad \text{(Capacity)}$$

$$\sum_{S\subseteq E} x_i(S) \leq 1 \qquad \forall i \in \mathscr{I} \ \ (\pi_i) \qquad \text{(XOR)}$$

$$x_i(S) \in \{0,1\} \qquad \forall i \in \mathscr{I}, S \subset E \qquad \text{(Binary)}$$

In the objective function, we aim to maximize the sum of valuations for assigned routes, i.e. the social welfare. Here, we also assume that a driver might have requests for two or more different routes. In an online version, this might be different requests, arriving in a sequence. The Capacity constraint enforces the respective capacity limit of road segment e for multiple trips a driver wants to take within the relevant period of time. The model uses the XOR bidding language, i.e. assigns at most one route out of several possible ones to a driver (Constraint XOR). Dual variables can be found to the right in brackets. The dual of the relaxed Primal with $0 \leq x_i(S) \leq 1$ for all $i \in \mathscr{I}$ and $S \subseteq E$ can now be formulated as follows:

$$\min \sum_{e \in E} c(e)p(e) + \sum_{i \in \mathscr{I}} \pi_i \tag{Dual}$$

subject to

$$\sum_{e \in E} a_i p(e) + \pi_i \geq v_i(S) \qquad \forall S \subset E, i \in \mathscr{I} \tag{Payoff}$$

$$p(e), \pi_i \geq 0 \qquad \forall e \in E, \forall i \in \mathscr{I} \tag{Real}$$

Corresponding to the m capacity constraints for all roads in the primal problem, we now have dual variables $p = (p(1), \cdots, p(m))$ for all roads, and payoffs $\pi = (\pi_1, \cdots, \pi_n)$ for all drivers. The sum of the dual prices on a route serves as a threshold price for a specific route. From LP duality we know that if (p^*, π^*) is an optimal solution for the Dual, then the primal optimal solution must satisfy

$$x_i(S) = \begin{cases} 1, & v_i(S) > a_i^T p^* \\ 0, & v_i(S) < a_i^T p^* \end{cases}$$

where $a_i = (a(1)_i, \cdots, a(m)_i)$. Thus, the prices p^* serve as thresholds in the mechanism such that only drivers with a willingness-to-pay higher than the (sum of) the shadow prices for all roads on a route get a route assigned. Note that the Dual can be rewritten as

$$f(p) = \min \sum_{e \in E} c(e)p(e) + \sum_{i \in \mathscr{I}} \left(v_i(S) - \sum_{e \in E} a_i p(e) \right)^+$$
$$p(e) \geq 0, e = 1, \ldots, m$$

where $(\cdot)^+$ is a ReLu function. This convex objective function can be interpreted as an n-sample approximation of a stochastic programming problem (Li et al. 2020).

In the offline version of the relaxed Primal problem (assuming trips were divisible), we would get competitive equilibrium prices such that each driver maximizes his payoff π_i at the prices $p(e)$ (Bikhchandani and Mamer 1997). However, our problem has integer variables and it needs to be solved online with drivers arriving over time.

4 Online Optimization

In our algorithm, we slice the day into time slots (e.g., 30 minutes) for which we can predict the number n of vehicles (or trips between origin-destination pairs) with high accuracy shortly before the time slot starts. Within this time slot, we use an online algorithm to adapt prices to changing demand in real-time. For this, we leverage a simple and fast subgradient descent algorithm on $f(p)$ (Li et al. 2020). This algorithm allows us to update prices continually. The algorithm performs one pass of projected stochastic subgradient descent in the dual space and determines the primal solution based on the dual iterate solution online. Note that a bidder's index i is determined solely based on the ordering of arrival. In what follows, we will thus replace the subscript $i \in \mathscr{I}$ by $t \in \mathscr{I}$, in order to emphasize that each request of a driver comes at a specific time, one after the other.

This online algorithm starts at the prices from the last period $p_{\text{last period}}$, observes the input requests or bids of the Primal sequentially, and computes the value of decision variable $x_t(S)$ after each request $(v_t(S), a_t)$. At each time t, it updates the current price vector p_t at step t with the new observation $(v_t(S), a_t)$. In Algorithm 1, Line 4, the gradient step is performed.

The price will decrease for roads that were either not requested or when the request was rejected as both of these result in $a_t x_t(S) = 0$ at the corresponding positions. Only if the request contains road e and was accepted, the price will increase by $\gamma(1 - c/n)$ since typically $c << n$ and thus $c/n < 1$. In Algorithm 1, Line 5, the price is projected to the feasible region of (non-negative) prices. The operator $\vee$ describes the element-wise maximum of both vectors, and γ_t is the step-size. We use a step size of $\gamma_t = 1/\sqrt{n}$, for which the regret and constraint violations are $O(m\sqrt{n})$ (Li et al. 2020). The resulting solution is an approximation to the optimal solution of the Primal.

Data: Input: $c, n, p_{\text{last period}}$

1 Initialize $p_1 = p_{\text{last period}}$

2 for $t = 1, \cdots, n$ do

3 $\quad x_t(S) = \begin{cases} 1, & v_t(S) > a_t^T p_t \\ 0, & v_t(S) \leq a_t^T p_t \end{cases}$

4 $\quad p_{t+1} = p_t + \gamma_t \left(a_t x_t(S) - (c/n)\right)$

5 $\quad p_{t+1} = p_{t+1} \vee 0$

6 end

7 return $x = (x_1(S), \cdots, x_n(S))$

Algorithm 1: Simple Online Algorithm

Note that the algorithm requires only one single pass through the input and is free of any matrix inversion or other costly operation. The algorithm directly outputs an integer solution to the relaxed linear program, an approximate solution to the integer linear program.

In order to avoid exceeding capacity constraints with non-stationary arrivals of requests, pricing should incorporate how much capacity is remaining for a specific road and raise prices accordingly. The modification of Algorithm 1 used in our experiments works similar to the simple online algorithm, but takes residual capacities into account during the price update. More precisely, Algorithm 1, Line 4 is replaced by the following:

$c_t = c_{t-1} - a_t x_t(S)$
$p_{t+1} = p_t + \gamma_t \left(a_t x_t(S) - (c_t/(n-t))\right)$

This replacement has two main features. Firstly, if the remaining capacity for a road e approaches 0, the term $c_t/(n-t)$ in the respective price update approaches 0, too. This leads to higher subsequent price updates whenever an additional capacity is sold and inversely, the price decreases at a slower rate whenever the road is not requested or the request is rejected. Secondly, the denominator now reflects the number of still to be expected requests. While in the beginning of a time slot this has little implications, a road with a lot of remaining capacity would get substantially cheaper after the majority of requests has been processed since the term $n-t$ approaches 0 then, leading to a greater value $c_t/(n-t)$. This allows for more adaptive admission of trips, which is reflecting the current traffic situation better.

5 Experimental Environment

In our experimental evaluation, we draw on SimMobility as a widely used traffic simulator (Adnan et al. 2016). It integrates state-of-the-art scalable simulators to predict the impact of mobility demands on transportation networks, intelligent transportation services, and vehicular emissions. The platform enables the simulation of the effects of a portfolio of technology, policy and investment options under alternative scenarios.

SimMobility is an activity-based, multi-modal simulation for different levels of temporal abstraction. It offers three distinct levels of simulation granularity: In the long-term framework, agents are simulated regarding land use and economic activity such as changing their workplace or their residence (year-to-year). The mid-term simulates transportation demands based on an agent's activity and travel patterns (day-to-day). Vehicles are moved on an aggregated level. Finally, the short-term framework provides a detailed simulation of individual vehicles. This includes acceleration, stopping at red lights, and other microscopic movements. We use the mid-term framework and the provided prototypical city to evaluate the effect of real-time road pricing on aggregate movements during a full day of traffic. The provided city has around 100,000 households and 254 roads segments, and features different types of roads, distinct households, and establishments as well as public transport networks. Available modes of transportation include private (car, motorbike, bus, taxi) and public (bus, subway) options while the purpose of a trip is either work or leisure. The parameters of the framework are set such that congestion arises.

In SimMobility, one can distinguish two types of agents: private individuals and professional drivers. Agents in the first group have an individual activity schedule including pre-determined start times, trip purpose and preferred modes of transportation which they follow for the duration of the simulation. Whenever such an agent starts a trip, she tries to find a connection between her origin and destination pair using the pre-determined mode from the activity schedule. If the agent prefers a mode of public transport they use a combination of walking, waiting, bus and MRT (a metro system) legs to arrive at their destination. Since prices for public transport are constant and known before the simulation starts, agents intending to use public transport can always do so and simply choose the fastest route via public transportation.

On the other hand, roads are priced dynamically and typically several routes connecting origin and destination in the road network are available. If the agent intends to use a private vehicle, she thus needs to incorporate both costs and travel times in her route choice. To this end, the agent uses a quasi-linear utility function $u_t(S) = v_t(S) - (p_t(S) + 10 * (tt_t(S) - (tt_t(Q_S)))$ to rank alternatives where $p_t(S)$ denotes the congestion price and $10 * (tt_t(S) - tt_t(Q_S)$ the costs for expected additional travel time tt_t (in hours) for path S at time t relative to the minimal travel time $tt_t(Q_S)$ of all paths between OD at time t. The valuation $v_t(S)$ for the trip is drawn from the uniform distribution over $[5, 15]$. If at least one route results in a utility $u_t(S) > 0$, the agent chooses the route with the highest utility and informs the service provider about the intention to drive on this path. The agent does not need to communicate her true willingness-to-pay $v_t(S)$ but only make a choice for one of the routes between origin and destination. While prices are valid for a short amount of time (e.g., 30 seconds), a trip can also be rejected by the service provider in case a driver waits too long and the prices change (e.g., due to an accident).

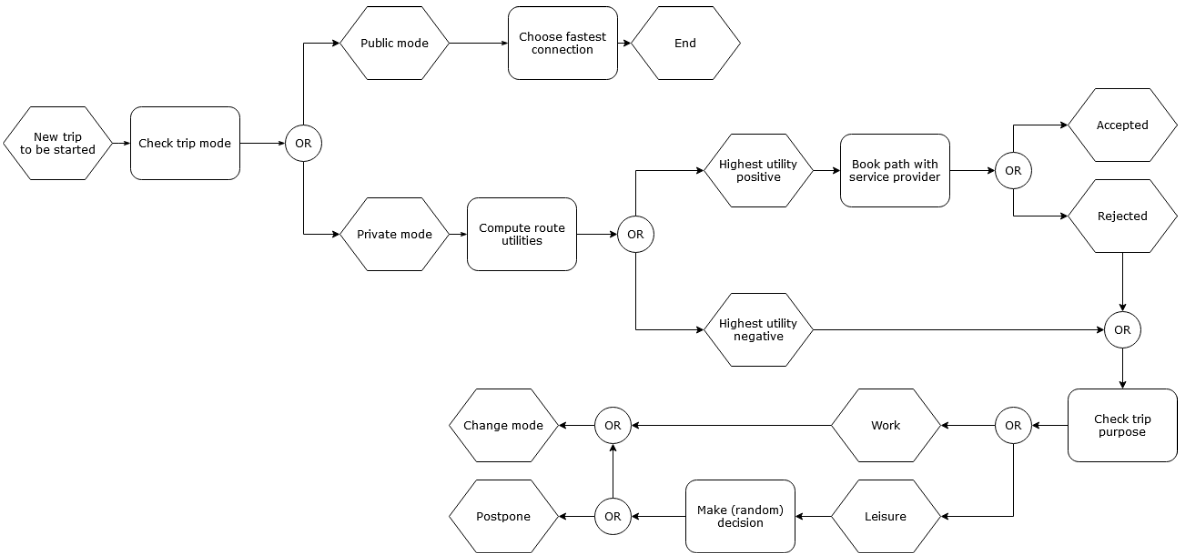


Figure 1: Individual agents' mode and route choice model

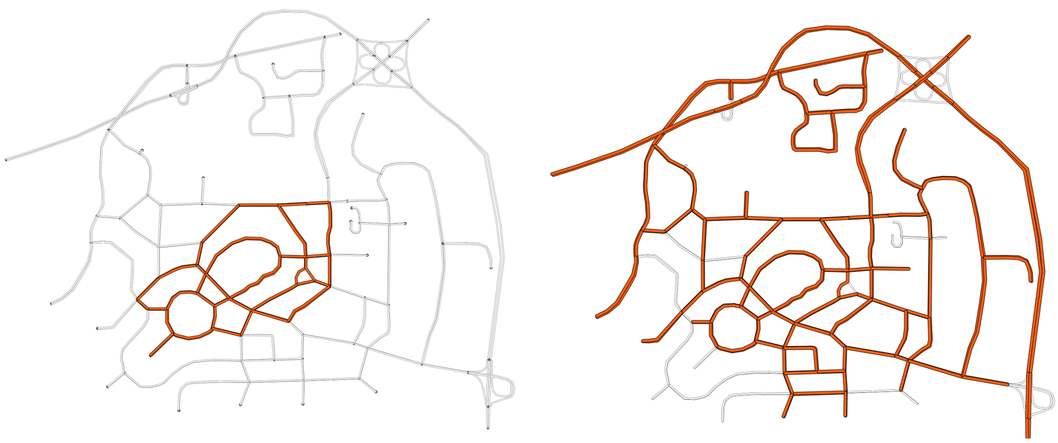
If, however, all routes result in a negative utility the agent refrains from booking a route with the service provider and instead either changes to public transport or postpones the trip once by 30 minutes. Modes of leisure trips are changed with a probability of 30% (the remaining 70% are postponed) while work trips cannot be postponed and always change modes. In case the mode of a trip was changed, the agent also changes subsequent schedule items if the intention was to use the same, now left behind, private vehicle. The overall process for this group of agents is depicted in Figure 1.

The second group in the simulation constitutes professional drivers such as taxi drivers. These agents have no pre-determined activity schedule, but instead decide where to drive during run-time. We assume that these drivers are on subscription plans so while they still book paths with the service provider to indicate the use of road capacity they will not be affected by dynamic prices. Professional drivers stick to their mode of transportation while all other drivers can switch to public transit.

6 Results

In our simulations, we analyze the overall number of traffic jams on a given day, the total welfare considering the true values of drivers, the revenue of the road service provider, and the number of realized private mode trips. We define a road segment to be congested if the average density in a timeslot is $\geq \frac{185 \text{ cars}}{\text{lane km}}$ (Knoop and Daamen 2017). Welfare is the sum of valuations $v_t(S)$ of completed trips, while revenue is the sum of the prices paid by drivers.

We consider three different mechanisms: static (zonal) pricing, dynamic pricing on the level of road segments, no pricing. For static prices, we defined a zone and charged either a low or a high fixed price whenever a driver enters the zone on his trip (see Figure 2a). The key parameter in our online algorithm is the number of trips requests we expect in a time period (30 minutes). We ran simulations with different levels of predictive accuracy, which we do not report due to page restrictions. For the reported results, we used the number of trips in the activity schedule per time slot as a forecast, which is not very precise. It does not include the trips of professional drivers as these are only determined at run-time (around 10-15% of overall traffic) and also it does not take into account delayed trips. In total, this leads to significant deviations from the actual number of trips requested in each time period. In spite of this poor forecast, the real-time pricing can still reduce the number of traffic jams effectively and achieve very high welfare. In this sense, the numbers we report are very conservative for the benefits that can be achieved with real-time dynamic pricing.



(a) Priced zone in static pricing (bold)

(b) Express and urban roads priced in real-time pricing (bold)

Figure 2: Road network and priced road segments

The average results of a simulation of one day (3 repetitions) are summarized in Table 1. Each repetition takes roughly three hours with SimMobility. When road usage is free of charge, we achieve a social welfare of 1.0 million $ on average and a high number of traffic jams. 76,218 planned trips with private transport modes can be completed. If drivers are

prices	# jams	std. dev.	$ welfare	std. dev.	$ revenue	std. dev.	realized priv. trips	std. dev.
none (baseline)	2,111	27	1,002,940	4,383	0	0	76,218	2,778
static (low)	1,938	56	953,811	5,425	205,352	1,957	70,407	2,729
static (high)	854	30	1,299,298	7,336	261,010	2,797	77,046	1,409
real-time	784	50	1,396,007	11,788	432,001	14,063	109,457	5,199

Table 1: Results for 4 different pricing mechanisms and standard deviations

stuck in traffic for too long, they cannot conduct a second or third trip on that day, which leads to a reduction in the number of actual trips. This serves as our baseline scenario.

In the low price scenario, we charge drivers for a payment of $7 in case they want to enter the priced zone. The results show that this is not an effective measure to prevent jams since most of the drivers are willing to pay this price and only around 21,000 people on average deviate to public transport. The number of traffic jams is still relatively high (Table 1). Yet, the small share of drivers changing to public transport leads to an even lower number of completed private trips than the baseline. The mechanism does, however, transfer part of the social welfare from the drivers to the service provider which earns around $205,000. In other words, low static prices reduce the drivers' utility (an average of $2.92 per trip) without reducing traffic jams.

The high static price scenario sets the price to $13. This discourages an average of 95,500 of the private travelers from using their cars or motorbikes and reduces the number of jams greatly. Only few drivers still want to afford driving into the priced area, but they pay a high price, leading to a revenue of around $261,000 for only 77,000 realized private trips. Note that this number of completed trips is higher than in the low-price scenario, because in the latter many drivers cannot complete their trips due to traffic congestion. With high prices of $13, drivers can largely use uncongested roads. Private drivers who use their private vehicles typically buy paths around the priced area (74%) and only a small fraction affords the luxury to use road segments in the priced area, resulting in a average price pf $3.39 per trip. The welfare with high static prices is relatively high, because most drivers leave their cars at home and take public transit.

Finally, real-time pricing controls prices for a number of road segments in the network, and effectively guides traffic to routes with less frequently used road segments. In this scenario, we price only express and urban roads (Figure 2b) and reserve 20% of the capacity for professional drivers. This achieves the highest welfare for the travelers ($1.39 million) and prevents almost two out of three traffic jams compared to the unpriced baseline scenario (with 2111 jams). In contrast to the high static prices, this approach sets incentives to use uncongested roads and leads to the completion of a high number of private trips (109,457). Additionally, on average 90,000 drivers opt to use public transit. Overall, the price signals effectively guide drivers to use the least congested routes and thus makes effective use of the existing road capacity. In contrast, in the treatment with high static prices most private drivers just use public transportation and the road capacity is underutilized. In our simulation, we assume that public transportation has sufficient capacity, which might not always be the case. If there is enough capacity of public transportation facilities and there is a goal to set incentives for drivers to use public transit for environmental reasons, the regulator can always increase the real-time dynamic price.

Prices depend on the corresponding requests and one might assume high prices follow a high number of requests and vice versa. Figure 3 exemplarily pictures the price development of road segment 88. Requests may or may not include a specific road segment and are either accepted or rejected. Out of the four combinations, only requests that include a certain road segment and are accepted increase the price of this road segment while requests that are either rejected or do not include the road segment lead to a price reduction in our gradient descent algorithm. The ratio between these groups of requests thus plays a crucial role. Note that if the total number of requests n is very high relative to the requests for a specific road segment, then prices can even go down because the road segment is relatively more attractive to other road segments in the traffic network in this price update rule. We have worked with different price update rules to analyze their results on welfare and the reduction of traffic jams. A more comprehensive discussion with different price update rules can be found in the long version of the paper.

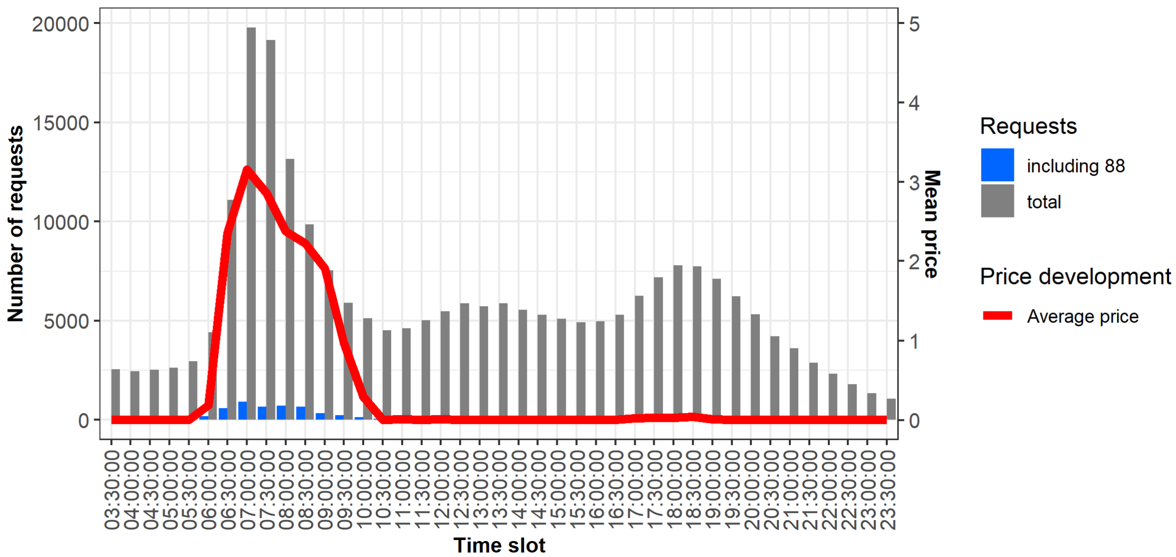


Figure 3: Price development of road segment 88 (solid line) compared to the number of requests for this road segment (histogram)

7 Conclusions

Traffic congestion is the key problem in the future of mobility. Current mechanisms to solve this problem by pricing roads include cordon-based, zonal-based and distance-based pricing. Unfortunately, such static pricing is known to be inefficient. On the other hand, real-time, trip- and distance-based pricing has only received little attention due to its inherent complexity. We suggest a system where drivers can buy either long-term plans for re-occurring trips (e.g., commutes) or short-term road capacity for individual trips. We focus on pricing of the latter. Our approach is based on short-term traffic forecasting and recent trends in online optimization that allow for real-time price updates of individual road segments in real-time.

Using the state-of-the-art traffic simulation framework SimMobility, we show that these real-time dynamic pricing mechanisms are computationally feasible, and that they achieve substantially higher welfare while reducing the number of jams compared to static pricing. Also, this approach does not discourage road usage in general (as is the case with high static prices), but instead indicates which roads are approaching their capacity limits and guides drivers effectively to pick routes which are less congested. Therefore, real-time dynamic pricing makes best use of scarce road capacity. Our approach also preserves privacy as drivers do not need to reveal their true willingness-to-pay. Drivers only need to reveal whether they are willing to accept a price or not.

The reported results are based on a simulation with poor forecasts of the number of requests expected in a time period (the only parameter needed in our algorithm) and yet real-time pricing outperforms static pricing methods by far. With better forecasting, we expect to reduce traffic jams even more effectively. Overall, there is a lot of room for fine tuning. However, the benefits of real-time dynamic pricing were already obvious in a wide range of simulation settings that we tested.

Acknowledgements

This project was supported by the Deutsche Forschungsgemeinschaft (DFG) (GRK 2201).

References

Adnan, Muhammad, Francisco C Pereira, Carlos M L Azevedo, Kakali Basak, Milan Lovric, Sebastián Raveau, Yi Zhu, Joseph Ferreira, Christopher Zegras, Moshe Ben-Akiva. 2016. Simmobility: A multi-scale integrated agent-based simulation platform. 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record.

Agrawal, Shipra, Zizhuo Wang, Yinyu Ye. 2014. A dynamic near-optimal algorithm for online linear programming. Operations Research 62(4) 876–890.

Bichler, Martin. 2017. Market design: a linear programming approach to auctions and matching. Cambridge University Press.

Bikhchandani, Sushil, Sven de Vries, James Schummer, Rakesh V Vohra. 2011. An ascending vickrey auction for selling bases of a matroid. Operations research 59(2) 400–413.

Bikhchandani, Sushil, John W Mamer. 1997. Competitive equilibrium in an exchange economy with indivisibilities. Journal of economic theory 74(2) 385–413.

Cheng, Qixiu, Zhiyuan Liu, Feifei Liu, Ruo Jia. 2017. Urban dynamic congestion pricing: an overview and emerging research needs. International Journal of Urban Sciences 21(sup1) 3–18.

Chiu, Yi-Chang, Jon Bottom, Michael Mahut, Alexander Paz, Ramachandran Balakrishna, Travis Waller, Jim Hicks. 2011. Dynamic traffic assignment: A primer. Dynamic Traffic Assignment: A Primer .

de Palma, André, Robin Lindsey. 2011. Traffic congestion pricing methodologies and technologies. Transportation Research Part C: Emerging Technologies 19(6) 1377–1399.

de Palma, André, Nathalie Picard. 2005. Route choice decision under travel time uncertainty. Transportation Research Part A: Policy and Practice 39(4) 295–324.

Do Chung, Byung, Tao Yao, Terry L Friesz, Hongcheng Liu. 2012. Dynamic congestion pricing with demand uncertainty: A robust optimization approach. Transportation Research Part B: Methodological 46(10) 1504–1518.

Hensher, David A. 2018. Tackling road congestion–what might it look like in the future under a collaborative and connected mobility model? Transport policy 66 A1–A8.

Knoop, Victor L, Winnie Daamen. 2017. Automatic fitting procedure for the fundamental diagram. Transportmetrica B: Transport Dynamics 5(2) 129–144.

Laval, Jorge A, Hyun W Cho, Juan C Muñoz, Yafeng Yin. 2015. Real-time congestion pricing strategies for toll facilities. Transportation Research Part B: Methodological 71 19–31.

Li, Xiaocheng, Chunlin Sun, Yinyu Ye. 2020. Simple and fast algorithm for binary integer and online linear programming. arXiv preprint arXiv:2003.02513 .

Li, Xiaocheng, Yinyu Ye. 2019. Online linear programming: Dual convergence, new algorithms, and regret bounds. arXiv preprint arXiv:1909.05499 .

Lo, Hong K, Wai Yuen Szeto. 2004. Modeling advanced traveler information services: static versus dynamic paradigms. Transportation Research Part B: Methodological 38(6) 495–515.

Rouhani, Omid M, Debbie Niemeier. 2014. Resolving the property right of transportation emissions through public–private partnerships. Transportation Research Part D: Transport and Environment 31 48–60.

Vickrey, William S. 1963. Pricing in urban and suburban transport. The American Economic Review 53(2) 452–465.

Wang, Jian, Libing Chi, Xiaowei Hu, Hongfei Zhou. 2014. Urban traffic congestion pricing model with the consideration of carbon emissions cost. Sustainability 6(2) 676–691.

Wie, Byung-Wook, Roger L Tobin. 1998. Dynamic congestion pricing models for general traffic networks. Transportation Research Part B: Methodological 32(5) 313–327.

6 Discussion and Conclusion

While the Vickrey-Clarke-Groves (VCG) mechanism is known to be the only efficient, strategy-proof mechanism, it is rarely used in practice (Rothkopf, 2007). Ausubel et al. (2006) attribute this phenomenon to low seller revenues and vulnerability to coordinated deviation by a coalition of bidders. Furthermore, the VCG mechanism requires repeated solving of the underlying allocation problem to optimality. There is, however, a great number of problems where no polynomial time algorithm is known due to their combinatorial nature. Algorithms which find an approximate solution can be deployed, but the VCG mechanism loses its properties if the allocation problem is not solved to optimality.

In this dissertation, we suggest pricing mechanisms for settings where the VCG mechanism falls short. We consider applications where the allocation needs to be approximated, budget constraints need to be considered, or bids arrive in an online fashion, and provide algorithms appropriate for the practical requirements. Despite their non-convex structure, instances of relevant size can be solved for all problems in feasible time, and we use prices to achieve important goals such as strategyproofness or core-stability.

As a first example, we consider a network procurement problem which needs to be approximated in practical applications (Chapter 3). In this setting, a telecommunications provider aims to connect a crucial subset of the available nodes at minimal cost. The provider needs to buy connections from suppliers which have private information about the cost of the connections they offer. We suggest a procurement auction to reduce transaction cost and increase competition. The underlying allocation problem refers to the Steiner minimum tree (SMT) problem, which belongs to Karp's 21 NP-complete problems (Karp, 1972), and hence does not have a polynomial time approximation within 96/95 unless P=NP (Chlebík and Chlebíková, 2008). A number of approximations for the SMT problem is available, but, prior to our work, the strategyproofness of mechanisms based on these approximations has been barely examined (Gualà and Proietti, 2005). Based on the assumption of single-minded bidders, we extend approximation algorithms to strategyproof mechanisms for network procurement.

Besides approximations for the SMT problem, we applied deferred acceptance auctions (DAA). The DAA is a general greedy-out approach that is even weakly group-strategyproof, but, for most problems, including the SMT, no worst-case approximation ratios are known. Our analysis shows that the DAA with simple scoring functions finds better solutions than approximations specifically designed for the SMT problem on average. Moreover, the premium for weak group-strategyproofness is surprisingly low. The convincing DAA performance is arguably the main result of this project. Among the monotonicity assessments, the disproof of monotonicity of the algorithm by Robins and Zelikovsky stands out because the complex algorithm defied intuition regarding monotonicity of its allocation. Moreover, the algorithm has the best known approximation ratio expect for Byrka et al. (2010), where de-randomization leads to a blow-up in the polynomial terms but computes allocations which are on average inferior to those computed by DAAs. The results give additional evidence for the strong performance of DAAs and provide valuable insights for the design of strategyproof network procurement auctions beyond the VCG mechanism.

Being restricted to the quasi-linear setting is another deficiency of the VCG mechanism. Markets where participants are constrained by budgets lie outside the quasi-linear setting and are thus unsuitable for the VCG mechanism. Such budget constraints are a valid concern in the secondary market for airport time slots we consider in our second project (Chapter 4). Since strategyproofness cannot be achieved under budget constraints (Dobzinski et al., 2012), we aim for core stable outcomes instead. The corresponding problem of computing a welfare-maximizing core-stable outcome in combinatorial exchanges with financially constrained bidders is in the complexity class of Σ_2^{P} (Bichler and Waldherr, 2019). Problems in this complexity class are often considered intractable, and only few solution strategies exist.

We use a mixed-integer bilevel linear program formulation of the problem and apply an algorithm by Zeng and An (2014). We improve the performance of this general purpose algorithm by leveraging problem-specific constraints, which enables us to solve instances of relevant size for the secondary airport time slots market. In a comparison to the initial implementation, we demonstrate the evident benefits of using problem-specific constraints regarding reduced runtimes. Moreover, we make two arguments in the favor of budget constraints. Firstly, when buyers are financially constrained but forced to state capped bids, the resulting outcome might actually be unstable. Furthermore, there is a considerable loss in welfare when budgets are ignored. In conclusion, we show that budget constraints need to be considered when setting prices in combinatorial exchanges. Despite the complexity of the resulting problem, we can solve instances of practically relevant size in a non-quasi-linear setting where the VCG mechanism is inapplicable.

The online setting is another example where the VCG mechanism falls short. As optimal allocations are generally unattainable in online markets, one of the central assumptions in the VCG mechanism is violated. Consequently, welfare losses are inevitable in these markets because decisions need to be made on incomplete information. Yet, there are applications where a timely decision based on partial information is preferable over a delayed decision. Congestion pricing, which we consider in our third project, is an important representative of online markets (Chapter 5). In this problem, road prices need to be set such that traffic congestion is reduced. Real-time congestion prices smoothly adapt road prices according to the current traffic situation, providing more flexibility than non-responsive pricing schemes. However, this requires nearly instant price updates as many requests need to be processed in real-time. We use a fast algorithm based on online linear programming (Li et al., 2020) and assess its performance on the SimMobility framework, a state-of-the-art traffic simulation. We show that real-time prices can reduce the number of congestions and increase the social welfare of travelers more than static area-based prices can. In comparison to the unpriced scenario, about two out of three traffic jams can be prevented, greatly increasing the social welfare of travelers. Our experiments further demonstrate the capability of the algorithm by Li et al. to compute prices quickly enough to satisfy practical requirements in a domain where the VCG mechanism is inadequate.

In conclusion, the assumptions in the VCG mechanism are not correctly reflecting all requirements in practice. Our results show that the shortcomings of the VCG mechanism in three of these markets can be overcome by dedicated pricing mechanisms.

References

Adnan, M., Pereira, F.C., Azevedo, C.M.L., Basak, K., Lovric, M., Raveau, S., Zhu, Y., Ferreira, J., Zegras, C., Ben-Akiva, M., 2016. Simmobility: A multi-scale integrated agent-based simulation platform, in: 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record.

Agrawal, S., Wang, Z., Ye, Y., 2014. A dynamic near-optimal algorithm for online linear programming. Operations Research 62, 876–890.

Andrew, L., Barman, S., Ligett, K., Lin, M., Meyerson, A., Roytman, A., Wierman, A., 2013. A tale of two metrics: Simultaneous bounds on competitiveness and regret, in: Conference on Learning Theory, pp. 741–763.

Ausubel, L.M., Milgrom, P., et al., 2006. The lovely but lonely vickrey auction. Combinatorial auctions 17, 22–26.

Avenali, A., D'Alfonso, T., Leporelli, C., Matteucci, G., Nastasi, A., Reverberi, P., 2015. An incentive pricing mechanism for efficient airport slot allocation in europe. Journal of Air Transport Management 42, 27–36.

Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R., 2008. Online auctions and generalized secretary problems. ACM SIGecom Exchanges 7, 1–11.

Balseiro, S.R., Besbes, O., Weintraub, G.Y., 2015. Repeated auctions with budgets in ad exchanges: Approximations and design. Management Science 61, 864–884.

Bard, J.F., 2013. Practical bilevel optimization: algorithms and applications. volume 30. Springer Science & Business Media.

Berman, P., Ramaiyer, V., 1994. Improved approximations for the steiner tree problem. Journal of Algorithms 17, 381–408.

Bichler, M., 2017. Market Design: A Linear Programming Approach to Auctions and Matching. Cambridge University Press, Cambridge. doi:10.1017/9781316779873.

Bichler, M., Goeree, J., Mayer, S., Shabalin, P., 2014. Spectrum auction design: Simple auctions for complex sales. Telecommunications Policy 38, 613–622.

Bichler, M., Goeree, J.K., 2017. Handbook of spectrum auction design. Cambridge University Press, Cambridge. doi:10.1017/9781316471609.

Bichler, M., Hao, Z., Littmann, R., Waldherr, S., 2020a. Strategyproof auction mechanisms for network procurement. OR Spectrum 42, 965–994. doi:10.1007/s00291-020-00597-7.

Bichler, M., Littmann, R., 2020. Real-time dynamic congestion pricing: An online optimization approach, in: Workshop on Information Technology and Systems (WITS20).

Bichler, M., Littmann, R., Waldherr, S., 2019. Combinatorial exchanges for airport time slots, in: Workshop on Information Technology and Systems (WITS19).

Bichler, M., Littmann, R., Waldherr, S., 2020b. Incentive-compatible auction mechanisms for network procurement, in: Proceedings of the 15th International Conference on Wirtschaftsinformatik. doi:10.30844/wi_2020_c4-bichler.

Bichler, M., Merting, S., 2018. Truthfulness in advertising? approximation mechanisms for knapsack bidders. European Journal of Operational Research 270, 775–783.

Bichler, M., Milgrom, P., Schwarz, G., 2020c. Taming the communication and computation complexity of combinatorial auctions: The fuel bid language URL: `https://milgrom.people.stanford.edu/sites/g/files/sbiybj4391/f/taming_the_communication_and_computation_complexity_of_combinatorial_auctions_the_fuel_bid_language_14-mar-2020.pdf`.

Bichler, M., Schneider, S., Guler, K., Sayal, M., 2011. Compact bidding languages and supplier selection for markets with economies of scale and scope. European Journal of Operational Research 214, 67–77.

Bichler, M., Waldherr, S., 2019. Computing core-stable outcomes in combinatorial exchanges with financially constrained bidders., in: EC, p. 747.

Bikhchandani, S., Ostroy, J.M., 2002. The package assignment model. Journal of Economic theory 107, 377–406.

Blumrosen, L., Nisan, N., 2007. Combinatorial auctions. Algorithmic game theory 267, 300.

Börgers, T., Krahmer, D., 2015. An introduction to the theory of mechanism design. Oxford University Press, USA.

Borodin, A., El-Yaniv, R., 2005. Online computation and competitive analysis. Cambridge University Press.

Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.

Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L., 2010. An improved lp-based approximation for steiner tree, in: Proceedings of the Forty-second ACM Symposium on Theory of Computing, ACM, New York, NY, USA. pp. 583–592. URL: `http://doi.acm.org/10.1145/1806689.1806769`, doi:10.1145/1806689.1806769.

Cheng, Q., Liu, Z., Liu, F., Jia, R., 2017. Urban dynamic congestion pricing: an overview and emerging research needs. International Journal of Urban Sciences 21, 3–18.

Cheng, X., Du, D.Z., 2013. Steiner trees in industry. volume 11. Springer Science & Business Media.

Chin, A.T., 1996. Containing air pollution and traffic congestion: transport policy and the environment in singapore. Atmospheric Environment 30, 787–801.

Chlebík, M., Chlebíková, J., 2008. The steiner tree problem on graphs: Inapproximability results. Theoretical Computer Science 406, 207–214.

Chopra, S., Gorres, E.R., Rao, M., 1992. Solving the steiner tree problem on a graph using branch and cut. ORSA Journal on Computing 4, 320–335.

Clarke, E., 1971. Multipart pricing of public goods. Public Choice XI, 17–33.

Corolli, L., Lulli, G., Ntaimo, L., 2014. The time slot allocation problem under uncertain capacity. Transportation Research Part C: Emerging Technologies 46, 16–29.

Coté, A., Meyerson, A., Poplawski, L., 2008. Randomized k-server on hierarchical binary trees, in: Proceedings of the fortieth annual ACM symposium on Theory of computing, pp. 227–234.

Cramton, P., 2010. How best to auction natural resources, in: Daniel, P., Keen, M., McPherson, C. (Eds.), The Taxation of Petroleum and Minerals. Routledge, London and New York, pp. 289–316.

Cramton, P., 2013. Spectrum auction design. Review of industrial organization 42, 161–190.

Cramton, P., Geddes, R.R., Ockenfels, A., 2017. Markets in road use: Eliminating congestion through scheduling, routing, and real-time road pricing. Working Paper .

Cramton, P.C., Shoham, Y., Steinberg, R., et al., 2006. Combinatorial auctions. volume 475. MIT press Cambridge.

Currie, J., Walker, R., 2011. Traffic congestion and infant health: Evidence from e-zpass. American Economic Journal: Applied Economics 3, 65–90.

Day, R., Milgrom, P., 2008. Core-selecting package auctions. International Journal of Game Theory 36, 393–407. URL: `https://doi.org/10.1007/s00182-007-0100-7`, doi:10.1007/s00182-007-0100-7.

Day, R., Raghavan, S., 2007. Fair payments for efficient allocations in public sector combinatorial auctions. Management Science 53, 1389–1406.

Day, R.W., Cramton, P., 2012. Quadratic Core-Selecting Payment Rules for Combinatorial Auctions. Operations Research 60, 588–603. URL: `https://pubsonline.informs.org/doi/abs/10.1287/opre.1110.1024`, doi:10.1287/opre.1110.1024.

Dempe, S., 2002. Foundations of bilevel programming. Springer Science & Business Media.

Dempe, S., Franke, S., 2014. Solution algorithm for an optimistic linear stackelberg problem. Computers & Operations Research 41, 277–281.

Dobzinski, S., Lavi, R., Nisan, N., 2012. Multi-unit auctions with budget limits. Games and Economic Behavior 74, 486–503.

Dreyfus, S.E., Wagner, R.A., 1971. The steiner problem in graphs. Networks 1, 195–207.

Erickson, R.E., Monma, C.L., Veinott Jr, A.F., 1987. Send-and-split method for minimum-concave-cost network flows. Mathematics of Operations Research 12, 634–664.

Ewers, H.J., 2001. Possibilities for the better use of airport slots in germany and the eu. a practical approach. Fachgebiet für Wirtschafts-und Infrastrukturpolitik TU Berlin .

Fischetti, M., Ljubić, I., Monaci, M., Sinnl, M., 2017. A new general-purpose algorithm for mixed-integer bilevel linear programs. Operations Research .

Fu, H., Li, Z., Wu, C., Chu, X., 2014. Core-selecting auctions for dynamically allocating heterogeneous vms in cloud computing, in: 2014 IEEE 7th International Conference on Cloud Computing, IEEE. pp. 152–159.

Fuchs, B., Kern, W., Wang, X., 2007. Speeding up the dreyfus–wagner algorithm for minimum steiner trees. Mathematical methods of operations research 66, 117–125.

Gallego, G., Van Ryzin, G., 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. Management science 40, 999–1020.

Goeree, J.K., Lien, Y., 2016. On the impossibility of core-selecting auctions. Theoretical Economics 11, 41–52. URL: `https://onlinelibrary.wiley.com/doi/abs/10.3982/TE1198`, doi:10.3982/TE1198.

Goetzendorff, A., Bichler, M., Shabalin, P., Day, R.W., 2015. Compact bid languages and core pricing in large multi-item auctions. Management Science 61, 1684–1703.

Green, J., Laffont, J.J., 1977. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. Econometrica: Journal of the Econometric Society , 427–438.

Groves, T., 1973. Incentives in teams. Econometrica 41, 617–631.

Gualà, L., Proietti, G., 2005. A truthful (2 - 2/k)-approximation mechanism for the steiner tree problem with k terminals*, in: Proceedings of the 11th Annual International Conference on Computing and Combinatorics, Springer-Verlag, Berlin, Heidelberg. pp. 390–400. doi:10.1007/11533719_40.

Gupta, A., Molinaro, M., 2014. How experts can solve lps online, in: European Symposium on Algorithms, Springer. pp. 517–529.

Herbster, M., Warmuth, M.K., 1998. Tracking the best expert. Machine learning 32, 151–178.

Hougardy, S., Prömel, H.J., 1999. A 1.598 approximation algorithm for the steiner problem in graphs, in: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms, pp. 448–453.

Jeyakumar, V., Lasserre, J.B., Li, G., Phm, T., 2016. Convergent semidefinite programming relaxations for global bilevel polynomial optimization problems. SIAM Journal on Optimization 26, 753–780.

Karp, R., 1972. Reducibility among combinatorial problems, in: Miller, R., Thatcher, J. (Eds.), Complexity of Computer Computations. Plenum Press, pp. 85–103.

Karpinski, M., Zelikovsky, A., 1997. New approximation algorithms for the steiner tree problems. Journal of Combinatorial Optimization 1, 47–65.

Karush, W., 1939. Minima of functions of several variables with inequalities as side constraints. M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago .

Kesselheim, T., Tönnis, A., Radke, K., Vöcking, B., 2014. Primal beats dual on online packing lps in the random-order model, in: Proceedings of the forty-sixth annual ACM symposium on Theory of computing, pp. 303–312.

Kleniati, P.M., Adjiman, C.S., 2014. Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development. Journal of Global Optimization 60, 425–458.

Kleniati, P.M., Adjiman, C.S., 2015. A generalization of the branch-and-sandwich algorithm: from continuous to mixed-integer nonlinear bilevel problems. Computers & Chemical Engineering 72, 373–386.

Koch, T., Martin, A., 1998. Solving steiner tree problems in graphs to optimality. Networks: An International Journal 32, 207–232.

Koh, A., 2013. A metaheuristic framework for bi-level programming problems with multidisciplinary applications, in: Metaheuristics for Bi-level Optimization. Springer, pp. 153–187.

Krishna, V., 2009. Auction Theory. Academic press.

Kuhn, H.W., Tucker, A.W., 2014. Nonlinear programming, in: Traces and emergence of nonlinear programming. Springer, pp. 247–258.

Lahaie, S., Pennock, D.M., Amer-Yahia, S., Parkes, D., 2013. Method and apparatus for administering a bidding language for online advertising. US Patent 8,527,353.

Lehmann, D., Müller, R., Sandholm, T., 2006. The winner determination problem. Combinatorial auctions , 297–318.

Lehmann, D., Oćallaghan, L.I., Shoham, 2002. Truth revelation in approximately efficient combinatorial auctions. J. ACM 49, 577–602. URL: `http://doi.acm.org/10.1145/585265.585266`, doi:10.1145/585265.585266.

Li, X., Sun, C., Ye, Y., 2020. Simple and fast algorithm for binary integer and online linear programming. arXiv preprint arXiv:2003.02513 .

Li, X., Ye, Y., 2019. Online linear programming: Dual convergence, new algorithms, and regret bounds. arXiv preprint arXiv:1909.05499 .

Lin, M., Wierman, A., Andrew, L.L., Thereska, E., 2012. Dynamic right-sizing for power-proportional data centers. IEEE/ACM Transactions on Networking 21, 1378–1391.

Lou, Y., Yin, Y., Laval, J.A., 2011. Optimal dynamic pricing strategies for high-occupancy/toll lanes. Transportation Research Part C: Emerging Technologies 19, 64–74.

Ltd., D.E., 2001. Auctioning airport slots: A report for hm treasury and the department of the environment, transport, and the regions .

Lu, C.C., Zhou, X., 2014. Short-term highway traffic state prediction using structural state space models. Journal of Intelligent Transportation Systems 18, 309–322.

Lubin, B., Juda, A.I., Cavallo, R., Lahaie, S., Shneidman, J., Parkes, D.C., 2008. ICE: an expressive iterative combinatorial exchange. Journal Artificial Intelligence Research 33, 33–37.

Lucena, A., Beasley, J., 1998. A branch and cut algorithm for the steiner problem in graphs. Networks: An International Journal 31, 39–59.

MacDonald, M., 2006. Study on the impact of the introduction of secondary trading at community airports. Report commissioned by the European Commission .

Madas, M.A., Zografos, K.G., 2013. Implementation of airport demand management strategies: a european perspective. Modelling and Managing Airport Performance , 233.

Mehta, A., 2013. Online matching and ad allocation URL: `https://storage.googleapis.com/pub-tools-public-publication-data/pdf/9b9bacbb54799b5b7b38fb53aa81ccc3b55604db.pdf`.

Milgrom, P., 2004. Putting Auction Theory to Work. Churchill Lectures in Economics, Cambridge University Press, Cambridge. URL: `https://www.cambridge.org/core/books/putting-auction-theory-to-work/63FA2A2D332E9E3A3238B6D5B650F2A5`, doi:10.1017/CBO9780511813825.

Milgrom, P., Strulovici, B., 2009. Substitute goods, auctions, and equilibrium. Journal of Economic theory 144, 212–247.

Mitsos, A., Barton, P.I., 2006. Issues in the development of global optimization algorithms for bilevel programs with a nonconvex inner program. Submitted to: Journal of Global Optimization .

Mitsos, A., Lemonidis, P., Barton, P.I., 2008. Global solution of bilevel programs with a nonconvex inner program. Journal of Global Optimization 42, 475–513.

Mölle, D., Richter, S., Rossmanith, P., 2006. A faster algorithm for the steiner tree problem, in: Annual Symposium on Theoretical Aspects of Computer Science, Springer. pp. 561–570.

Mookherjee, D., 2008. The 2007 nobel memorial prize in mechanism design theory. The Scandinavian Journal of Economics , 237–260.

Myerson, R.B., 1981. Optimal auction design. Mathematics of operations research 6, 58–73.

Nisan, N., 2006. Bidding languages. Combinatorial auctions , 400–420.

Nisan, N., Bayer, J., Chandra, D., Franji, T., Gardner, R., Matias, Y., Rhodes, N., Seltzer, M., Tom, D., Varian, H., et al., 2009. Googles auction for tv ads, in: International Colloquium on Automata, Languages, and Programming, Springer. pp. 309–327.

Organization, I.I.C.A., 2016. Report of the economic commission on agenda item 39, report a39-wp/518, montreal, quebec, canada.

de Palma, A., Lindsey, R., 2011. Traffic congestion pricing methodologies and technologies. Transportation Research Part C: Emerging Technologies 19, 1377–1399.

Parkes, D.C., Kalagnanam, J., Eso, M., 2001. Achieving budget-balance with Vickrey-based payment schemes in exchanges, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence, pp. 1161–1168.

Paulavičius, R., Adjiman, C.S., 2019. An enhanced basbl solver for bilevel problems. Numerical Computations: Theory and Algorithms NUMTA 2019 , 219.

Paulavičius, R., Adjiman, C.S., 2020. New bounding schemes and algorithmic options for the branch-and-sandwich algorithm. Journal of Global Optimization , 1–29.

Paulavičius, R., Gao, J., Kleniati, P.M., Adjiman, C.S., 2020. Basbl: Branch-and-sandwich bilevel solver. implementation and computational study with the basblib test set. Computers & Chemical Engineering 132, 106609.

Paulavičius, R., Kleniati, P.M., Adjiman, C.S., 2016. Global optimization of nonconvex bilevel problems: implementation and computational study of the branch-and-sandwich algorithm, in: Computer Aided Chemical Engineering. Elsevier. volume 38, pp. 1977–1982.

Pellegrini, P., Bolić, T., Castelli, L., Pesenti, R., 2017. Sosta: An effective model for the simultaneous optimisation of airport slot allocation. Transportation Research Part E: Logistics and Transportation Review 99, 34–53.

Pellegrini, P., Castelli, L., Pesenti, R., 2012. Secondary trading of airport slots as a combinatorial exchange. Transportation Research Part E: Logistics and Transportation Review 48, 1009–1022.

Pigou, A.C., 1920. The economics of welfare. Palgrave Macmillan.

Prömel, H.J., Steger, A., 2000. A new approximation algorithm for the steiner tree problem with performance ratio 5/3. Journal of Algorithms 36, 89–101.

Rajesh, J., Gupta, K., Kusumakar, H.S., Jayaraman, V.K., Kulkarni, B.D., 2003. A tabu search based approach for solving a class of bilevel programming problems in chemical engineering. Journal of Heuristics 9, 307–319.

Roberts, D.J., Postlewaite, A., 1976. The incentives for price-taking behavior in large exchange economies. Econometrica: Journal of the Econometric Society , 115–127.

Robins, G., Zelikovsky, A., 2005. Tighter bounds for graph steiner tree approximation. SIAM J. Discret. Math. 19, 122–134. doi:10.1137/S0895480101393155.

Rotaris, L., Danielis, R., Marcucci, E., Massiani, J., 2010. The urban road pricing scheme to curb pollution in milan, italy: Description, impacts and preliminary cost–benefit analysis assessment. Transportation Research Part A: Policy and Practice 44, 359–375.

Rothkopf, M.H., 2007. Thirteen reasons why the Vickrey-Clarke-Groves process is not practical. Operations Research 55, 191–197.

Simeonova, E., Currie, J., Nilsson, P., Walker, R., 2018. Congestion pricing, air pollution and childrens health. Technical Report. National Bureau of Economic Research.

Tahernejad, S., Ralphs, T.K., DeNegre, S.T., 2017. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. Working Paper .

Ulrich, C., 2008. How the present (iata) slot allocation works. Airport Slots: International Experiences and Options for Reform , 9–20.

Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance 16, 8–37.

Vickrey, W.S., 1952. The revision of the rapid transit fare structure of the city of New York. 8.

Voß, S., 2006. Steiner tree problems in telecommunications, in: Handbook of optimization in telecommunications. Springer, pp. 459–492.

Yin, Y., Lou, Y., 2009. Dynamic tolling strategies for managed lanes. Journal of Transportation Engineering 135, 45–52.

Zelikovsky, A., 1996. Better approximation bounds for the network and euclidean steiner tree problems. University of Virginia, Charlottesville, VA .

Zeng, B., An, Y., 2014. Solving bilevel mixed integer program by reformulations and decomposition. Optimization online , 1–34.

Zinkevich, M., 2003. Online convex programming and generalized infinitesimal gradient ascent, in: Proceedings of the 20th international conference on machine learning (icml-03), pp. 928–936.

Zografos, K.G., Salouras, Y., Madas, M.A., 2012. Dealing with the efficient allocation of scarce resources at congested airports. Transportation Research Part C: Emerging Technologies 21, 244–256.