FACULTY OF ECONOMICS AND
BUSINESS ADMINISTRATION

COMBINATORIAL AUCTIONS:

THEORY, PRACTICE, AND EXPERIMENTS

Bart Vangerven

Supervisors:
Prof. Dr. Dries R. Goossens, Prof. Dr. Frits C.R. Spieksma

Dissertation presented to obtain the degree of Doctor in
Business Economics at Ghent University and KU Leuven

Academic year: 2016 – 2017

GHENT
UNIVERSITY

Doctoral committee

Supervisors:

Prof. dr. Dries R. Goossens
Ghent University, Faculty of Economics and Business Administration

Prof. dr. Frits C.R. Spieksma
KU Leuven, Faculty of Economics and Business

Members:

Prof. dr. Martina Vandebroek
KU Leuven, Faculty of Economics and Business

Prof. dr. Tarik Aouam
Ghent University, Faculty of Economics and Business Administration

Prof. dr. Marion Ott
RWTH Aachen, School of Business and Economics

Acknowledgments

> Knowledge is in the end based on acknowledgement
>
> *Ludwig Wittgenstein*

The proverb ‘the longest mile is the last mile home’ quite accurately describes my feelings at the end of this PhD journey. As usual at the end of your PhD, there is one thing left to do to complete the doctoral dissertation: writing the only part of this behemoth that people might actually read — *the acknowledgements*. Acknowledgements are usually quite non-consequential, in the sense that people are rarely evaluated on them (unlike, hopefully, the rest of this work), yet they do matter. Amidst the celebration, the *right people* require thanks in the *right way*.

As is tradition, the first paragraph is dedicated to the advisor. Fun fact; while writing these acknowledgements, and going over how some of my predecessors handled it before me, I noticed that it is even tradition to mention that it is tradition to mention the advisor first. Fortunately, I had the privilege of having not just one, but two advisors. I would like to express my sincere gratitude to my advisors Prof. Dr. Frits Spieksma and Prof. Dr. Dries Goossens for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. Their door was - usually - open whenever I ran into a spot of trouble or had a question. Their guidance and attention to detail helped me bring my PhD journey to a good end. I could not have imagined having better advisors and mentors for my PhD study.

Besides my advisors, I would like to thank the rest of my doctoral committee: Prof. Dr. Martina Vandebroek, Prof. Dr. Tarik Aouam, and Prof. Dr. Marion Ott, for their insightful comments and encouragement, but also for the questions which helped improve this dissertation immensely.

PhD students often talk about loneliness during the course of their study. That is something which I never experienced. In fact, my time as a PhD student was characterized by many wonderful colleagues, who all deserve some manner of thanks, e.g. for all the stimulating discussions, (laughing at inappropriate) jokes, and for all the fun we have had. I will mention the three musketeers first: Bart Smeulders (II), Ward Passchyn (you put the "b" in subtle), and Annette Ficker (you are an excellent rubber duck). That conference and subsequent trip in Glasgow was amazing, and I am happy you introduced me to the world of board games.

Also on the 'fifth floor'. Starting with the OR of ORSTAT, I thank Kris Coolen (perhaps we were too much like a couple of fishwives), Daniel Kowalczyk (Łatwo przyszło, łatwo poszło), Guopeng Song & Fan Yang (hopefully someday I can pronounce your names correctly), Ben Hermans and Salim Rostami (brave enough to drink something that *smokes*). The fifth floor also houses the STAT part of ORSTAT: Viktoria Öllerer, Ines Wilms, Luca Barbaglia & Ruben Crevits. An observation: coffee and cake, or food in general, might be the secret to bringing OR and STAT (closer) together.

Moving on the the 'fourth floor', I thank Hamed Jalali (or would you rather I call you Prof. Dr. Hamed Jalali now?), Stef Lemmens (without that one heads up years ago this work would have been "Imposhibibble! Imposhibibble!"), Raïsa Carmen & Carla Van Riet (please continue running and motivating people!), Joren I & II (you decide which is which), Joeri Poppe and Morteza Davari (one day you will beat me in 'the only game that matters'), Michael Samudra (hopefully I convinced you to switch to the "good stuff"), Sarah

Van der Auweraer (check out the Nerf Super Soaker Zombie Strike Revenge Contaminator), Silvia Valeria Padilla Tinoco (pura vida!), and Sebastian '*speedy*' Gonzalez, Finally, the newcomers: Heletje Van Staden, Laurens Deprez, and Kim De Boeck. It falls on you to keep the cake traditions alive! Phew, I really hope I did not forget anyone in that list...

A lifelong friend obviously deserves mentioning. Tommy Graindourze, I like to think your philosophical ways kept me sane(r) during my PhD journey. The computer and board game distractions probably also helped.

Finally, this last paragraph is dedicated to family. Special thanks go to my sister, her husband and their lovely daughter, my (first) godchild. I also want to express my deep gratitude to my parents. Both have instilled many admirable qualities in me, and have given me a solid foundation in life. They provided me with unfailing love, support and continuous encouragement throughout my years of study and the process of researching and writing this thesis. This accomplishment would not have been possible without them. The clichéd 'words cannot express' is fitting. Still, I will try: mom and dad, I profoundly thank you.

Bart Vangerven
Leuven, December 2017.

English summary

This doctoral dissertation contributes to theory, experiments, and practice in combinatorial auctions. Combinatorial auctions are multi-object auctions, that enable bidders to bid on packages of items.

In Chapter 2, we theoretically investigate the classical winner determination problem in geometrical settings. Specifically, we consider combinatorial auctions of items that can be arranged in rows, and the objective is, given bids on subsets of items, to find a subset of bids that maximizes auction revenue. Possible practical applications include allocating pieces of land for real estate development, or seats in a theater or stadium. We investigate several geometrical structures and bid shapes, and provide either a polynomial dynamic programming algorithm or an NP-hardness proof, filling in several gaps in academic literature.

In Chapter 3, we combine theory with experiments, investigating coordination and threshold problems in combinatorial auctions. Bidders on small packages of items are unable to outbid provisionally winning bids on large packages alone; despite free-rider incentives, both coordination and cooperation are required. Coordination because smaller bidders need to bid on disjoint packages, and cooperation because more than one bidder is required to outbid a larger package bid. We propose indices quantifying both the coordination and the threshold problem, that can be used in providing feedback or generating valuations for laboratory experiments. Additionally, we develop coalitional feedback that is specifically

aimed at helping bidders to overcome coordination and threshold problems. We test this in an experimental setting using human bidders, varying feedback from provisionally winning bids and prices, to winning and deadness levels, and coalitional feedback. We find that in situations where threshold problems are severe, coalitional feedback increases economic efficiency, but in easy or insurmountable threshold problems that is not always the case.

Finally, in Chapter 4, we combine theory with practice. Scheduling a conference, based on preferences expressed by conference participants, can be seen as a combinatorial auction with public goods. In a situation with public goods, the utility of the final selected goods (presentations scheduled in parallel) are "consumed" by all bidders (conference participants). Constructing a good conference schedule is important: they are an essential part of academic research and require significant investments (e.g. time and money) from their participants. We provide computational complexity results, along with a combined approach of assigning talks to rooms and time slots, grouping talks into sessions, and deciding on an optimal itinerary for each participant. Our goal is to maximize attendance, considering the common practice of session hopping. On a secondary level, we accommodate presenters' availabilities. This personalized conference scheduling approach has been applied to construct the schedule of the MathSport (2013), MAPSP (2015 and 2017) and ORBEL (2017) conferences.

Nederlandse samenvatting

Dit doctoraal proefschrift draagt bij tot de theorie, experimenten, en praktijk van combinatorische veilingen. Combinatorische veilingen zijn veilingen met meerdere objecten, die bieders toelaten om pakketbiedingen te plaatsen.

In Hoofdstuk 2 onderzoeken we het klassieke winnaar determinatieprobleem in geometrische omgevingen. Meer specifiek, onderzoeken we combinatorische veilingen van objecten die in rijen gesorteerd zijn. De doelfunctie is dan om, gegeven biedingen op verzamelingen van objecten, de deelverzameling van objecten te selecteren die de opbrengst voor de veilingmeester maximaliseert. Potentiële praktische toepassingen zijn onder meer het toewijzen van stukken grond, of zitjes in een theater of stadion. We onderzoeken verschillende geometrische structuren en vormen van biedingen, en geven ofwel een polynomiaal dynamisch programmeringsalgoritme of een NP-moeilijkheidsbewijs. Op die manier vullen we verschillende gaten in de academische literatuur op.

In Hoofdstuk 3 combineren we theorie met experimenten, en onderzoeken we coördinatie- en drempelproblemen. Bieders op kleine pakketten van objecten kunnen voorlopig winnende biedingen op grote pakketten niet alleen verslaan; niettegenstaande vrijbuiters-incentieven, zijn zowel coördinatie als samenwerking nodig. Coördinatie omdat kleinere bieders op niet overlappende pakketten van objecten moeten bieden, en samenwerking omdat er meer dan één bieder nodig is om een groter pakketbod te overbieden. We stellen indices voor, die zowel het coördinatie- als het drempelprobleem kwantificeren, en die b.v. gebruikt kunnen worden om feedback aan bieders te voorzien of om waarderingen voor labo experimenten te genereren. Verder ontwikkelen we ook coalitiefeedback die specifiek gericht is om bieders te helpen coördinatie- en drempelproblemen te overkomen. We testen deze feedback in een experimentele omgeving met menselijke bieders, waar we feedback variëren van feedback over voorlopig winnende biedingen en prijzen, tot winning en deadness levels, en coalitiefeedback. We vinden dat in situaties met moeilijke drempelproblemen, coalitiefeedback economische efficiëntie verhoogt, maar dat in gemakkelijke of onoverkomelijke drempelproblemen dat niet altijd het geval is.

In Hoofdstuk 4 combineren we theorie met de praktijk. Het plannen van een conferentie gebaseerd op geuite voorkeuren van deelnemers, kan gezien worden als een combinatorische veiling met publieke goederen. In een situatie met publieke goederen, wordt het nut van de finaal geselecteerde goederen (presentaties in parallel gepland) genuttigd door alle bieders (deelnemers aan de conferentie). Een goed programmaschema opstellen is belangrijk, gezien conferenties een essentieel onderdeel van wetenschappelijk onderzoek vormen en significante investeringen (b.v. tijd en geld) vereisen van de deelnemers. We geven een NP-moeilijkheidsbewijs, samen met een alomvattende aanpak die presentaties aan lokalen en tijdstippen toewijst, die praatjes in sessies indeelt, en die een optimaal traject voor iedere deelnemer uitstippelt. Ons doel is om de aanwezigheid te maximaliseren, rekening houdend met mensen die tussen presentaties door naar andere sessies gaan. Op een tweede niveau, houden we rekening met de beschikbaarheden van de presentatoren. Deze gepersonaliseerde conferentieplanningsaanpak is gebruikt om de programmaschema's van MathSport (2013), MAPSP (2015 and 2017) and ORBEL (2017) op te stellen.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

> Begin - to begin is half the work,
> let half still remain; again begin
> this, and thou wilt have finished.
>
> *Marcus Aurelius*

1.1 About combinatorial auctions

This doctoral dissertation contributes to theory, experiments and practice in combinatorial auctions. Combinatorial auctions are a special type of auction, because they allow bidders to place bids on bundles (sometimes called combinations or packages) of items. This property enables bidders to more fully express their preferences. This is in stark contrast with the traditional (sequential) single item auctions. Logically, it follows that combinatorial auctions make most sense in environments where bidders might have complementary values, i.e. synergies, that arise from the bundling of items. An often used example to demonstrate complementarities is the following: a pair of shoes is worth more than the value of a single left shoe and a single right shoe. Both bidders and the auctioneer can benefit from combinatorial auctions; economic efficiency as well as auction revenue are often increased when bidders are allowed to enter package bids. Combinatorial auctions have a long list of (potential) practical applications that account for millions of dollars of revenues. Without a doubt, the most well-known application is that of allocating spectrum licenses (Jackson, 1976; McMillan, 1994; Banks et al., 2003; Plott and Salmon, 2004; Seifert and Ehrhart, 2005; Günlük et al., 2005; Brunner et al., 2010; Scheffel et al., 2012; Fox and Bajari, 2013; Bichler et al., 2013). Broadly speaking, spectrum auctions seek to allocate licences to transmit signals over specific bands of the electromagnetic spectrum. Spectrum resources being scarce, it is important to allocate the spectrum licenses efficiently, i.e. to parties that value them the most, preferably providing significant revenues for governments as well. However, there are other applications as well, ranging from the allocation of airport landing slots (Rassenti et al., 1982) or harbor time slots (Ignatius et al., 2014) (time slots that are closer together time-wise have a better synergy than time slots that are further apart time-wise), and the allocation of mineral/oil drilling rights (Cramton, 2007) (synergies can be found if mineral/drilling right for several adjacent plots of land are secured). It follows that combinatorial auctions are well researched. In fact, there has been a surge of research in the last two decades.

1.2 Dissertation structure

Combinatorial auctions bring with them a diverse set of challenges. The work described in the following chapters attempts to tackle some of those challenges. It is presented in such a way that they can be read as stand-alone work.

Chapter 2 contains theoretical research on winner determination problems. The winner determination problem is the following: given a number of bids in a combinatorial auction, determine the allocation of items to bidders that maximizes the auctioneer's revenue, such that every item is sold at most once. Usually, in the academic literature, special cases are investigated where the winner determination problem can be solved efficiently (i.e. in polynomial time), or can be approximated to some degree, see e.g. Rothkopf et al. (1998), Sandholm et al. (2005), Xia et al. (2005), Günlük et al. (2005), Babaioff and Blumrosen (2008), Mu'alem and Nisan (2008). Chapter 2 is concerned with solving the winner determination problem in a geometrical setting. Specifically, we consider auctions of items that can be arranged in rows. Examples of such a setting appear naturally in allocating pieces of land for real estate development, or seats in a theater or stadium, as well as oil/mineral rights. In chapter 2 the objective is, given bids on subsets of items, to find a subset of bids that maximizes auction revenue in a pay-as-bid auction (i.e. winning bidders pay the price of their bids). We describe a dynamic programming algorithm which for a problem with k rows and so-called *connected* and *gap-free* bids, solves the winner determination problem in polynomial time. In addition, we study the complexity for bids in a grid, complementing known results in the literature. We also study variants of the winner determination problem in specific geometrical settings. We provide a NP-hardness proof for the 2-row setting with gap-free bids. Finally, we end chapter 2 with an extension of the dynamic programming algorithm to solve the case where bidders submit connected, but not necessarily gap-free bids in a 2-row and a 3-row problem.

Chapter 3 is about bidder support and combines theory with experiments. Combinatorial auctions, by allowing bidders to enter package bids, introduce both coordination and threshold problems. Bidders on small packages of items are unable to individually outbid provisionally winning bids on large packages; despite free-rider incentives, both coordination and cooperation are required. Coordination because smaller bidders need to bid on disjoint packages that are present in an efficient outcome, and cooperation because more than one bidder is required to outbid larger package bids that stand in the way of such an efficient outcome.

We study such coordination and threshold problems in combinatorial auctions and propose measures quantifying both the coordination and threshold problems. This is a novel contribution to the auction literature, where the coordination and threshold problems are often not separately discussed, but mixed together.

Also in this chapter, we devise a type of feedback dubbed coalitional feedback. Although giant strides have been made in recent years in the field of bidder support, there remains an open question as to how effective bidder support, in the shape of feedback, can help overcome coordination and threshold problems. Coalitional feedback is specifically aimed at helping bidders to overcome coordination and threshold problems. It follows that such feedback can be valuable in supporting bidders in a combinatorial auction. Finally, to put the theory to the (experimental) test, we test different levels of feedback in an experimental setting using human bidders, varying feedback from very basic information about provisionally winning bids and corresponding prices, to more advanced concepts such as winning and deadness levels, and coalitional feedback.

In Chapter 4 we present both theory and practice about conference scheduling. Scientific conferences have become an essential part of academic research and require significant investments (e.g. time and money) from their participants. It falls upon the organizers to develop a schedule that allows the participants to attend the talks of their interest, making their conference experience a good one. Maximizing attendance is only possible once participant preferences are known. Constructing a conference schedule based on these preferences forms the theme of this chapter. One way to elicit preferences of conference goers, is to give them a list of all talks along with the question: “Dear participant, which talks would you like to attend?” Participants are free to reveal their preferences this way. Every participant then has the opportunity to check the talks he or she wants to attend, resulting in a binary preference vector with length equal to the number of talks in the conference.

However, all wells, when dug deep enough, lead to the same water source. Indeed, we can think of our conference scheduling approach as a special type of combinatorial auction. More concrete, we can model preference based conference scheduling as a combinatorial auction with public goods. Observe that in a traditional combinatorial auction, the utility - an economic term referring to the satisfaction from consuming a good or service - of the items is obtained privately. That is: the utility of an item is acquired exclusively by the bidder winning the item. In a situation with public goods, the utility of the final selected goods are consumed by all bidders. We can model preference based conference scheduling as a combinatorial auction with public goods as follows. Each participant in the conference corresponds to a bidder in the auction, and each combination of talks that can take place in parallel correspond to an item. The preference vector of the conference participants can be seen as the valuation that a bidder expresses for a particular combination of parallel talks. More precise, the number of missed attendances that follows from a bidder's preference vector for a tuple can be seen as the bidder's expressed cost for that tuple. Next, summing these costs over the bidders, we arrive at the cost-coefficient for a combination of parallel talks. The problem is then to select the combinations of parallel talks in such a way that the selected cost coefficient is minimized, subject to the constraint that every talk has to be in exactly one selected combination of parallel talks. Equivalently, the goal is to maximize total attendance, i.e. welfare. Summarizing, in contrast to a combinatorial auction involving private goods, in this combinatorial public goods auction (i) not every item will/can be sold (not every combination of talks will be selected), and (ii) the final product is the conference schedule, which corresponds to a selected set of combinations of talks in parallel enjoyed by all participants. In fact, the public good auction we describe here is a variation of the closely related Combinatorial Public Project Problem, which was introduced by Papadimitriou et al. (2008), and further studied by Schapira and Singer (2008), Buchfuhrer et al. (2010), Lucier et al. (2013), and Markakis and Telelis (2017). The Combinatorial Public Project Problem consists of selecting k out of m available items, while maximizing social welfare. Instead of putting a constraint on the number of selected items, we constrain the selected items such that every talk is selected exactly once.

Setting aside auction terminology, we present in Chapter 4 a combined approach of assigning talks to rooms and time slots, grouping talks into sessions, and deciding on an optimal itinerary for each participant. Our primary goal is to maximize attendance, taking into account the common practice of *session hopping*. Session hopping is the phenomenon of participants switchings sessions between consecutive talks, in order to attend a talk in another session. This is often perceived as disturbing, yet session hopping is regular practice; it makes sense to, after maximizing possible attendance, minimize session hopping. On a secondary level, we accommodate presenters' availabilities. Finally, if room capacities are a potential issue, we also discuss how to assign sessions to rooms. We present theoretical contributions regarding the computational complexity of maximizing attendance, and discuss a dynamic programming algorithm to calculate the number of session hops exactly, as well as a heuristic to approximate the number of session hops quickly. Finally, we turn theory into practice and apply our conference scheduling approach to construct the schedule of four medium-sized scientific conferences: MathSport (2013), MAPSP (2015 and 2017) and ORBEL (2017).

Chapter 2

Winner Determination Problems in Geometrical Combinatorial Auctions

> Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.
>
> *Edsger Wybe Dijkstra*

2.1 Introduction

In combinatorial auctions, bidders can place bids on combinations of items, called packages or bundles. Clearly, combinatorial auctions allow bidders to better express their preferences compared to the traditional auction formats, where bidders place bids on individual

This chapter is based on Vangerven et al. (2017b).

items. In particular, it makes sense to use a combinatorial auction when complementarities or substitution effects exist between different items.

Research on combinatorial auctions was triggered by applications such as the FCC spectrum auction (Jackson, 1976) and auctions for airport time slots (Rassenti et al., 1982). For an introduction to combinatorial auctions, we refer to the book edited by Cramton et al. (2006); for a survey of the literature, we refer to Abrache et al. (2007) and de Vries and Vohra (2003).

One important challenge within this domain is, given the bids, to decide which items should be allocated to which bidder, i.e., which bids to accept. In general, this *winner determination problem* is NP-hard (Van Hoesel and Müller, 2001), and does not allow good approximation results (Sandholm, 2002).

We discuss a combinatorial auction in a restricted topology. In this setting, an item corresponds to a rectangle, and all items are arranged in (a limited number of) rows, see Figure 2.1 for an example. Notice that the individual items (or rectangles) need not have the

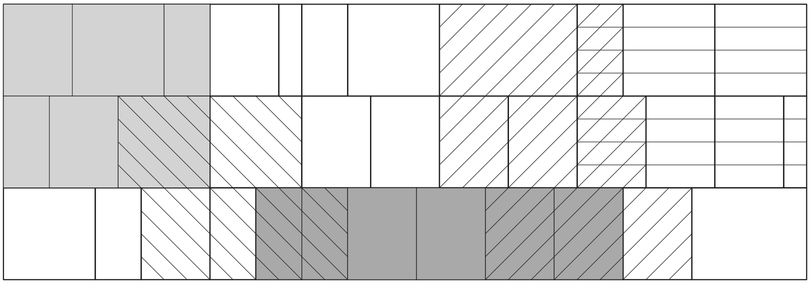


Figure 2.1: An example of an instance with 3 rows and 5 bids.

same size. A bid consists of a set of items satisfying some restrictions (see Section 2.2 for a precise problem definition), together with a value. The objective is to select a set of bids that maximizes the sum of the expressed values, while making sure that each item is present at most once in a selected bid.

There are several situations in practice that motivate this specific geometric setting. We mention the following:

- Real estate. Goossens et al. (2014) describe how space in a newly erected building, to be used for housing and commercial purposes, is allocated using a combinatorial auction. The geometric structure of each of the levels of the building features the properties described here. Quan (1994) reports on empirical studies in real estate auctions. Several of these studies have focused on verifying and quantifying the *afternoon effect*. This afternoon effect describes similar items consistently selling for significantly less in later rounds in multi-object sequential auctions. Quan (1994) even reports on finding this effect in a large real estate auction (122 lots) of vacant lots that are geographically similar. The lots were formed in 23 groups based on their geographical proximity. In 20 out of the 23 groups of properties, the afternoon effect was present with the last bidder paying on average one-third less than the first bidder for geographically similar lots. A combinatorial auction, by selling all items simultaneously, can mitigate this effect.

- Mineral rights. Imagine a region that is partitioned into lots, with the lots organized in rows. For sale is the right to extract minerals, oil or gas found on or below the surface of the lot. Clearly, having adjacent lots allows for exploration and production efficiencies, a complementarity. For more about this particular setting, we refer to Cramton (2007). Figure 2.2 shows an example of oil and gas leases neatly arranged in rows.

- Seats in a grandstand, theater or stadium. In some of these cases, one can even assume that a grid, consisting of rows and columns, is given where each cell represents a seat. Typically, demand exists for sets of adjacent seats - think of a family of four going to a ball game, or a group of friends visiting a concert. The complementarities that people perceive from adjacent seats offer possibilities for combinatorial auctions.

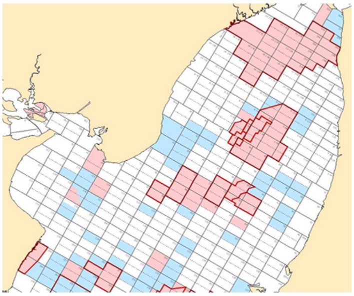


Figure 2.2: Oil and Gas Leases managed by the Texas General Land Office. Taken from: http://www.glo.texas.gov/GLO/agency-administration/gis/gis-data.html.

Although tickets are usually sold at a fixed price, there are occasions where sports teams have auctioned off (part of) their seat licenses[1]. Another, not unrealistic, example is the selling of airline tickets[2].

- Laboratory experiments. Scheffel et al. (2011) provide results of laboratory experiments testing different auction formats in five different value models. Their third value model has six pieces of land arranged in two rows on a shoreline. Bidders are interested in bundles that contain at least one lot at the shore. Their fourth value model has nine pieces of land arranged in three rows. In Scheffel et al. (2012) a local synergy value model is used in which 18 items are arranged rectangularly in three rows with bidders interested in adjacent items. Kazumori (2010) ran experiments using 16 items arranged rectangularly

[1] For instance, the New York Jets (NFL) have earned over $16 million in an online auction for seat licenses. See http://www.nfl.com/news/story/09000d5d80c071a4/article/jets-earn-more-than-16-million-in-online-psl-auction.

[2] For instance, the article found at the following URL describes how some carriers require persons whose weight exceeds a given number to buy two (adjacent) tickets: http://www.cheapair.com/blog/travel-tips/airline-policies-for-overweight-passengers-traveling-this-summer/.

> in four rows. Each agent has a base value for each item and a varying level of additional interest for adjacent items. These laboratory experiments required solving very small instances of the winner determination problem. In case one were to increase the number of pieces of land, or one wants to run a continuous auction, or one wants to give bidders all sorts of feedback, an efficient algorithm for the winner determination becomes a necessity.

In all these cases, it is clear that complementarities between adjacent items exist; a combinatorial auction is best-placed to take these effects into account.

The main goal of this chapter is to show how the specific geometric setting described above can be used to efficiently solve the winner determination problem (which is hard in general), using dynamic programming procedures. Additionally, we settle the complexity of the winner determination problem for bidding in a grid. This chapter does not address mechanism design or bidding strategy issues.

Goossens et al. (2014) show that when a constraint is imposed stating that a bidder can have at most one winning bid, the winner determination problem is NP-hard even if all items are arranged on a single row. Hence, to have any prospect of coming up with a positive result, we allow bidders to win multiple bids. Notice however that, under some conditions on the bids, an optimal solution where each bidder has at most one winning bid is guaranteed to exist. This is the case, for instance, if the bids placed by each bidder satisfy at least one of the following conditions:

- every pair of bids of a bidder has a non-empty intersection
- all bids from the same bidder are *super-additive*, i.e. for any two disjoint sets S and T it should hold that the bid expressed on $S \cup T$ is at least as large as the sum of the expressed bids on S and T.

The first condition is satisfied if bidders place only one bid. Bids coming from (truthful) *single-minded* bidders, who are only interested in a specific set of items or a superset of these items, also satisfy the first condition. Indeed, more formally, single-minded bidders have a set of items S^* and a value v^* such that their valuation $v(S) = v^*$ for all $S \supseteq S^*$, and $v(S) = 0$ for all other S (see Nisan et al. (2007)). The second condition corresponds to the bids that can be expressed using a bidding language consisting of OR-bids (see Nisan (2000)). Summarizing, in these cases, our dynamic program will result in an optimal solution where each bidder has at most one winning bid.

2.1.1 Literature

Our problem is a special case of finding a maximum-weight independent set in a geometric intersection graph. In such a graph, there is a node for each bid (in our case: a (connected) set of rectangles), and two nodes are connected if and only if the corresponding bids overlap. Finding a maximum-weight independent set in a geometric intersection graph is a well-studied problem for several types of intersection graphs. For instance, in the work of Rothkopf et al. (1998), it is shown that if all items are arranged in a single row, and bids are only allowed for subsets of consecutive items, the resulting winner determination problem is polynomially solvable. These results follow from the equivalence of this problem to finding a maximum-weight independent set in an interval graph. For an overview on results for more general intersection graphs we refer to Chan and Har-Peled (2012). Depending upon particular properties of the geometric figures, different complexity results are known. We restrict ourselves here to mentioning that for fat objects (like squares and disks) polynomial time approximation schemes are known (see Erlebach et al. (2001), Hochbaum and Maass (1985)). The important special case of finding a maximum-weight independent set in a rectangle intersection graph is considered in Chalermsook and Chuzhoy (2009).

In the context of auctions, Babaioff and Blumrosen (2008) and

Christodoulou et al. (2010) study mechanism designs for the setting where geometric figures in the plane are the objects for sale. They sketch applications in advertising, renting land for exhibitions and licenses for location-based services. They show how to guarantee a certain fraction of the optimal welfare for certain shapes of geometric objects (e.g. convex figures). The geometric setting considered here is different; also we do not devise payment schemes for the bidders. This chapter addresses the question of how to solve the winner determination problem, assuming bidders have placed bids for subsets of items.

Our problem is also somewhat related to rectangle packing. Given a set of rectangles, the rectangle packing problem is to find a bounding box (i.e. an enclosing rectangle) of minimum area that will contain the given rectangles without overlap. The optimization problem is NP-hard, while the problem of deciding whether a set of rectangles can be packed in a given bounding box is NP-complete (Leung et al., 1990). This resembles a setting where bidders want to acquire a set of seats in a theater, of given number and shape (e.g. four seats next to each other, a 3×2 block of seats, etc.), anywhere in the theater. This can be casted in our framework by having a bid for each possible set of seats. In general however, this problem is fundamentally different from our problem: in rectangle packing, given rectangles can be placed anywhere in the bounding box, whereas in our problem the position of the items are fixed and the decision to be made is whether or not to select a particular bid.

If we allow bidders to express multiple bids, the problem is NP-hard even in a setting where all items are arranged in a single row. Indeed, this follows immediately from the fact that the Job Interval Selection Problem (JISP) is MAX SNP-hard (Spieksma, 1999). In the JISP n pairs of intervals on the real line are given, and the objective is to select as many intervals as possible such that no two selected intervals intersect and at most one interval is selected from each pair.

2.1.2 Our Results

For the setting where items are arranged in rows, we show the following:

- For connected and gap-free bids (see Section 2.2 for precise definitions), the winner determination problem is easy when the number of rows is fixed (see Section 2.3.1). We provide a general polynomial time dynamic programming algorithm.

- For the setting where the bid space is a grid and both the number of rows and columns are a part of the input, we show that even when bids are constrained to be row bids or column bids, the resulting winner determination problem is NP-hard (see Section 2.3.3),

- For gap-free bids, the winner determination problem is NP-hard, even on two rows (see Section 2.4.1).

- For connected bids, the winner determination problem is easy on three rows or fewer (see Sections 2.4.2 and 2.4.3). We show this by adapting and expanding upon the general dynamic programming algorithm discussed in Section 2.3.1.

We point out that the complexity of the winner determination problem with connected bids on a fixed number of rows k, with $k \geq 4$, is still an open problem. If the number of rows is part of the input, a result in Rothkopf et al. (1998) implies the problem is NP-hard. An overview can be found in Table 2.1.

2.2 Preliminaries

The geometric setting that we consider can be described as follows. Given are k rows. Each row contains an (ordered) set of items (or rectangles). If, on some row, an item a lies to the left of item b, then we write $a \prec b$. We use $X_j = \{0, 1, \ldots, m_j\}$ to denote the set of items in row j, $j = 1, \ldots, k$. The set of items that can be bid on is

Rows	Connected and gap-free bids	Connected bids	Gap-free bids
1	$O(m+n)$		
2	$O(m^2+nm)$		NP-hard (Vialette, 2004)
3	$O(m^3+nm^2)$	$O(n^2m^3)$	
$k : k \geq 4$	$O(m^k+nm^{k-1})$	Open problem	

Table 2.1: Overview of results if the number of rows k is not part of the input, m is the number of items and n is the number of bids.

$\bigcup_{j=1}^{k} X_j \setminus \{0\}$; item 0 cannot be part of any bid, and is only present for notational convenience. We assume that item ℓ lies directly to the left of item $\ell+1$, for each $\ell \in X_j \setminus \{m_j\}$, $j = 1, \ldots, k$.

Definition 2.1. *We say that a pair of items are adjacent if and only if they share a border with non-zero length.*

Clearly, items ℓ and $\ell+1$ are adjacent. However, items on different (but consecutive) rows can be adjacent as well. We use m to denote the number of items in the instance, i.e., $m = \sum_{j=1}^{k} m_j$. Figure 2.3 visualizes this.

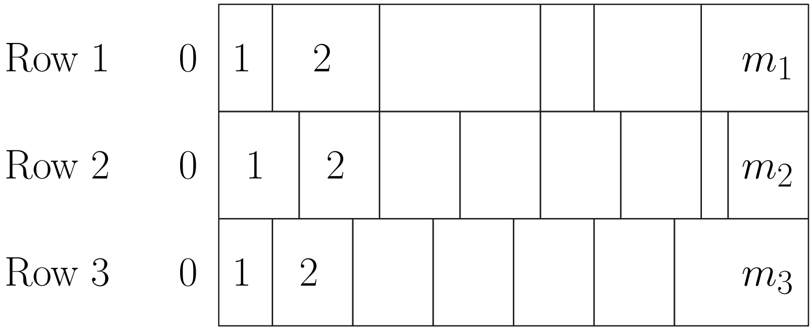


Figure 2.3: An example of an instance with $k = 3$ (i.e. 3 rows) and $m_1 = 6$, $m_2 = 8$, $m_3 = 7$.

We investigate the following problem, called the winner determination problem (WDP). Given is a set of bids $\mathcal{B}$ on subsets of items, with $v(b)$ denoting the value of bid b, for each $b \in \mathcal{B}$. We set $n = |\mathcal{B}|$, i.e. there are n bids; specifying a bid implies specifying a set of items, as well as a value $v(b) > 0$. The problem is to find an allocation that maximizes the sum of the values of the accepted bids, ensuring that each item is allocated at most once.

Given a bid b, consider the *item graph*, $H(b)$, which has a node for each item in bid b, and there is an edge between a pair of nodes in $H(b)$ if and only if the corresponding items are adjacent. There are two main restrictions on the bids that we consider. We define the concept of a *connected* bid.

Definition 2.2. *We say that bid b is* connected *if the subgraph $H(b)$ induced by the items of bid b is connected. If bid b is not connected, we say that it is* disconnected.

Further, let us define the concept of a bid that is *gap-free*. A formal definition of a bid having no gaps (i.e. being gap-free) is formulated as follows.

Definition 2.3. *We say that bid b is* gap-free *if no three items $u \prec v \prec w$ on a single row exist for which $u \in b, v \notin b, w \in b$.*

A bid that is not gap-free has at least one gap. Notice that it is easy to exhibit examples of connected bids that are not gap-free (see Figure 2.4), and gap-free bids that are not connected (see Figure 2.5). It is also easy to see that in the case of a single row, i.e. $k = 1$, connectedness of a bid is equivalent to a bid being gap-free.

Finally, it is important to see that bids on identical sets of items but with different values need not all be considered. Indeed, one need only consider the bid with the highest value. If more than one bid has the highest value, one could use the bid entry time as a tie-breaker. Thus, all but the highest value bid on a specific set of items can be eliminated and bids will be unique in the sense that they are all for different sets of items.

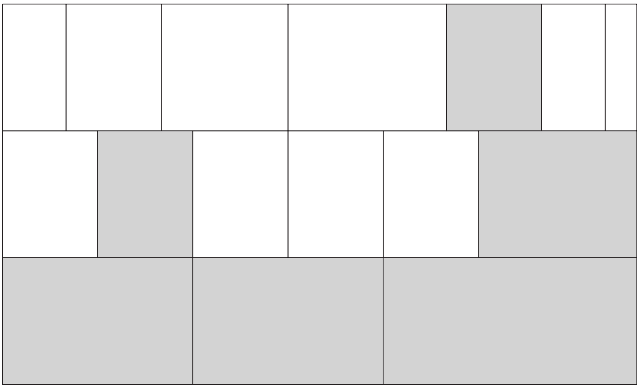


Figure 2.4: A bid that is connected and not gap-free.

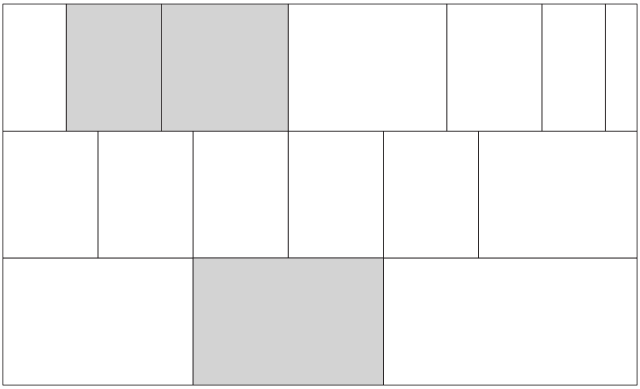


Figure 2.5: A bid that is disconnected and gap-free.

2.3 Connected and gap-free bids

In this section we assume that bids are connected and gap-free. In Section 2.3.1 we describe a dynamic programming algorithm for the winner determination problem, tailored to our geometric setting described in Section 2.2, for the general case of k rows. This algorithm has a polynomial running time if we assume that k is not part of the input, or in other words, if we focus on problem instances with a specific number of rows. Notice that this assumption is reasonable for practical applications, as the auctioneer will typically be interested in a setting with one particular number of rows, namely that number resulting from the specific geometric structure underlying the items for sale.

In Section 2.3.3, we abandon this assumption, and study a setting where problem instances can have any number of rows (i.e. the number of rows is part of the input). We discuss a setting where items are arranged in a grid, and show that this problem is difficult, even when bids can cover only items in one row or one column.

2.3.1 A dynamic program for winner determination for case of k rows

This section is divided as follows: first we describe the dynamic programming algorithm for the case of k rows, then we proceed to a numerical example, after which we will discuss the proof of correctness.

The dynamic program for k rows

Here we describe a dynamic programming approach for the case of k rows and bids that are both connected and gap-free. We show how the winner determination problem for this setting can be solved as a shortest path problem on a graph $G = (V, A)$, which is constructed as follows. There is a node in V for each element in the Cartesian product of the sets X_1,X_2, ..., X_k. We write $V = \prod_{i=1}^{k} X_i$. Nodes in V are k-tuples. We consider the k-tuple $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_k \rangle$, where $x_1 \in X_1$, $x_2 \in X_2$, $\ldots$ and $x_k \in X_k$. This k-tuple represents a state, i.e. a collection of assigned items. More specifically, the k-tuple $\boldsymbol{x}$ represents a state where irrevocable decisions concerning the items $\{0, \ldots, x_1\} \cup \{0, \ldots, x_2\} \cup \cdots \cup \{0, \ldots, x_k\}$ have been made, i.e. for each row i all items from left to right up to and including x_i. As there is a node in V for every k-tuple, this leads to $O(m^k)$ nodes.

The arc set A includes two types of arcs: the zero arcs and bid arcs. The zero arcs have a weight of 0, and are used to handle items not included in the set of winning bids. Consider some node $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_i, \ldots, x_k \rangle \in V$, with $1 \leq i \leq k$ and $x_i \neq m_i$. A zero arc goes from node $\boldsymbol{x}$ to node $\langle x_1, \ldots, x_i + 1, \ldots, x_k \rangle \in V$, for each $1 \leq i \leq k$. Thus, up to k zero arcs emanate node $\boldsymbol{x} \in V$, giving rise to $O(m^k)$ zero arcs in the graph G.

The bid arcs correspond to actual bids and have a weight equal to the value of the bid $v(b)$. We represent a bid by listing k pairs of elements; each pair represents the first element, and the last element present in a bid on a particular row. For a bid b that contains elements from each of the k rows, we write: $b = \{(x_1^b, y_1^b), (x_2^b, y_2^b), \ldots, (x_k^b, y_k^b)\}$, where the element $x_j^b \in X_j$ $(1 \leq j \leq k)$ refers to the leftmost element of X_j present in bid b, and the element $y_j^b \in X_j$ $(1 \leq j \leq k)$ refers to the rightmost element of X_j present in bid b. We use the symbol $(\varnothing, \varnothing)$ to denote that a bid does not include items from that row. Thus, as an example, when we write $b = \{(\varnothing, \varnothing), (x_2^b, y_2^b), (x_3^b, y_3^b), (\varnothing, \varnothing)\}$ this means that the bid b does not include any items on the first row, it includes items x_2 up to and including y_2 on the second row, it includes items x_3 up to and including y_3 on the third row, and it does not include any items on the fourth row.

The bid arcs can be described as follows. Let us, for convenience, first assume that bid b contains elements from each of the k rows. To represent bid b in the graph G, we draw an arc from node $\langle x_1^b - 1, x_2^b - 1, \ldots, x_k^b - 1 \rangle$ to node $\langle y_1^b, y_2^b, \ldots, y_k^b \rangle$ with weight $v(b)$. Consider now a bid b such that there are rows with no elements in b. Observe that, due to connectedness of b, these rows can only have indices $1, 2, \ldots, s(b)$ and $f(b), f(b)+1, \ldots, k$ with $0 \leq s(b) < f(b) \leq k+1$. Note that if a bid b is present on the row 1 then $s(b) = 0$. Similarly, if a bid b is present on row k then $f(b) = k+1$. Now, to represent bid b, for each $x_1 \in X_1$, $x_2 \in X_2$, $\ldots$, $x_{s(b)} \in X_{s(b)}$, $x_{f(b)} \in X_{f(b)}$, $x_{f(b)+1} \in X_{f(b)+1}$, $\ldots$, $x_k \in X_k$ there is an arc from node $\langle x_1, x_2, \ldots, x_{s(b)}, x^b_{s(b)+1} - 1, \ldots, x^b_{f(b)-1} - 1, x_{f(b)}, \ldots, x_k \rangle$ to node $\langle x_1, x_2, \ldots, x_{s(b)}, y^b_{s(b)+1}, \ldots, y^b_{f(b)-1}, x_{f(b)}, \ldots, x_k \rangle$ with weight $v(b)$. Notice that there are $O(nm^{k-1})$ bid arcs (of course it is conceivable that the number of bid arcs will be far less).

We now compute a longest path from node $\mathbf{0} = \langle 0, \ldots, 0 \rangle$ to node $\boldsymbol{m} = \langle m_1, \ldots, m_k \rangle$. The length of this path corresponds to the optimal revenue of the auction, and the winning bids can be derived from the arcs in the path. Notice that $G = (V, A)$ is acyclic by construction and consists of $O(m^k)$ nodes and $O(m^{k-1}(n+m))$ arcs. Hence, a longest path can be found efficiently by solving a shortest path problem in $G = (V, A)$ with edge weights multiplied by -1. Since Ahuja et al. (1993) show that shortest path problems in directed acyclic graphs with p nodes and q arcs can be solved in $O(p+q)$ time, our dynamic program requires $O(m^k + nm^{k-1})$ time.

Once a longest path is found, it is easy to see which bids are accepted. Every arc that is not a zero arc in $G = (V, A)$ corresponds to exactly one bid. To find the set of winning bids, for every non-zero arc in the longest path simply accept the bid corresponding to that arc.

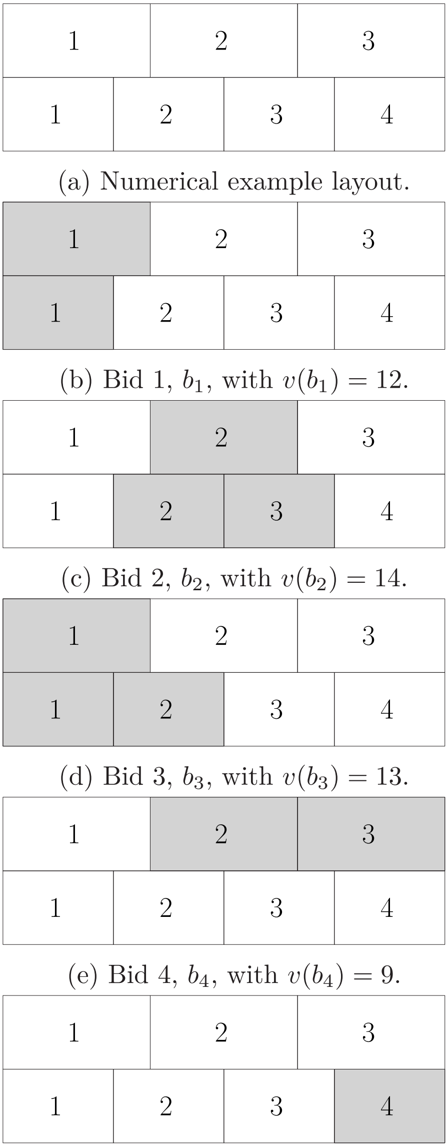
Note that when the shortest path problem in $G = (V, A)$ is solved, it is easy to get *reduced costs* (i.e. *shadow prices*) for all arcs. The minimum of the reduced costs of all arcs corresponding to a single bid, is the amount by which the bid needs to be improved (i.e. increased) to be winning, if all other bids remain the same. This amount by which a bid needs to be increased, ceteris paribus, to become a winning bid has been termed the *winning level* of a bid (Adomavicius and Gupta, 2005). Thus, in other words, the winning levels of currently non-winning bids are easy to compute by using our approach. This fact can be useful for providing feedback to bidders, helping bidders to better evaluate whether they should revise previous bids (Adomavicius et al., 2012).

A numerical example

In this section we will use the layout found in Figure 2.6a to illustrate how the graph $G = (V, A)$ is created. As can be seen in Figure 2.6a, the example has two rows. In the first row there are three items and in the second row there are four items. Five bids are submitted, see Figures 2.6b-2.6f. Bid 1, b_1 with $v(b_1) = 12$, is on the first item in row 1 and on the first item in row 2. Bid 2, b_2 with $v(b_2) = 14$, is on item 2 in row 1 and items 2 and 3 in row 2. Bid 3, b_3 with $v(b_3) = 13$, is on the first item in row 1 and the first two items in row 2. Bid 4, b_4 with $v(b_4) = 9$, is on the last two items in row 1. Bid 5, b_5 with $v(b_5) = 4$, is on the last item in row 2. Given these bids, now the winner determination problem needs to be solved.

Now let us construct the graph $G = (V, A)$. First we construct the nodes. There is a node in V for each element in $X_1 \times X_2$. In our example: $X_1 = \{0, 1, 2, 3\}$ and $X_2 = \{0, 1, 2, 3, 4\}$. The resulting nodes are 2-tuples or pairs, each of which represent a state: the first component denotes which items have already been handled on the first row, the second component denotes which items have already been assigned on the second row. For example, in the pair $\langle 1, 3\rangle$ we have already made decision regarding item 1 in row 1 and items 1, 2 and 3 in row 2. The appropriate nodes for the numerical example can be found in Figure 2.7.

Bid 1 can be represented as $b_1 = \{(1, 1), (1, 1)\}$. That means that there is a single arc corresponding to bid 1 from $\langle 0, 0\rangle$ to $\langle 1, 1\rangle$. Bid 2 can be represented as $b_2 = \{(2, 2), (2, 3)\}$. That means that there is an arc corresponding to bid 2 from $\langle 1, 1\rangle$ to $\langle 2, 3\rangle$. Bid 3 can be represented as $b_3 = \{(1, 1), (1, 2)\}$. That means that there is an arc corresponding to bid 3 from $\langle 0, 0\rangle$ to $\langle 1, 2\rangle$. Bid 4 can be represented as $b_4 = \{(2, 3), (\varnothing, \varnothing)\}$. This leads to multiple arcs: from $\langle 1, 0\rangle$ to $\langle 3, 0\rangle$, from $\langle 1, 1\rangle$ to $\langle 3, 1\rangle$, from $\langle 1, 2\rangle$ to $\langle 3, 2\rangle$, from $\langle 1, 3\rangle$ to $\langle 3, 3\rangle$, and from $\langle 1, 4\rangle$ to $\langle 3, 4\rangle$. Bid 5 can be represented as $b_5 = \{(\varnothing, \varnothing), (4, 4)\}$. This also leads to multiple arcs: from $\langle 0, 3\rangle$



(a) Numerical example layout.

(b) Bid 1, b_1, with $v(b_1) = 12$.

(c) Bid 2, b_2, with $v(b_2) = 14$.

(d) Bid 3, b_3, with $v(b_3) = 13$.

(e) Bid 4, b_4, with $v(b_4) = 9$.

(f) Bid 5, b_5, with $v(b_5) = 4$.

Figure 2.6: The layout of the items, five bids.

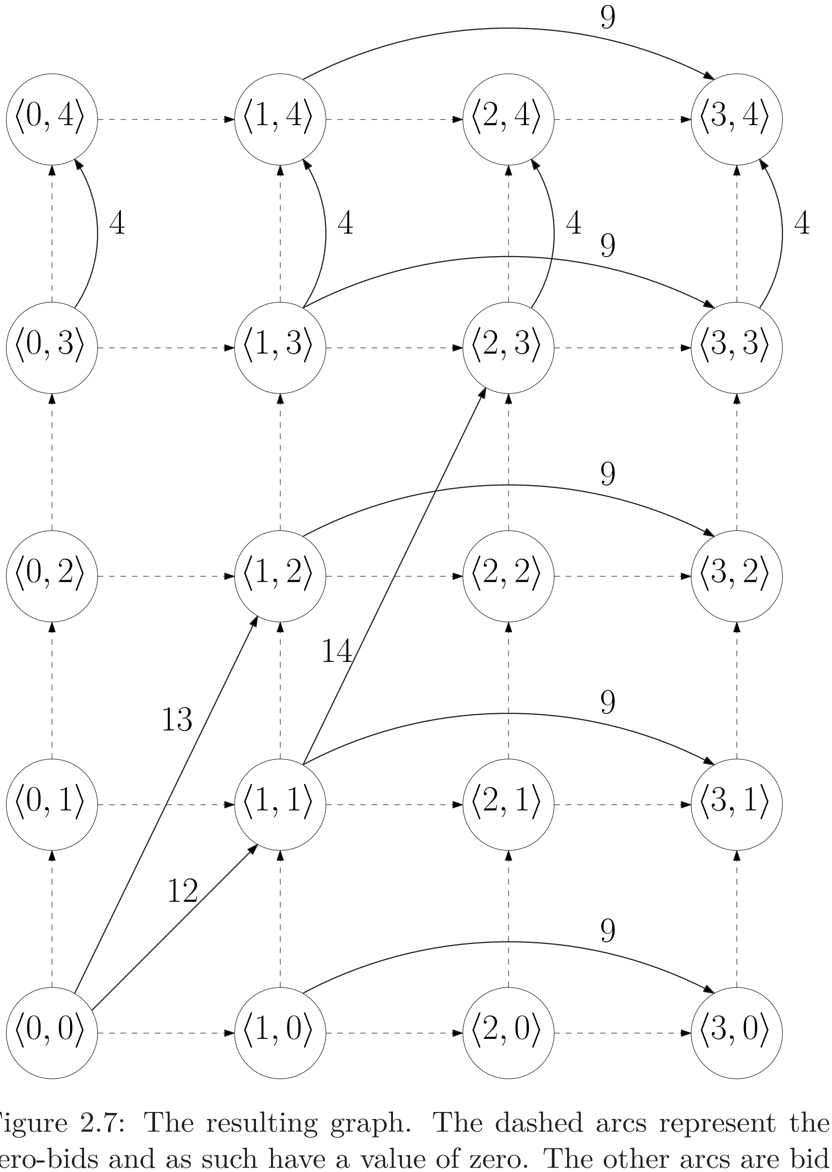


Figure 2.7: The resulting graph. The dashed arcs represent the zero-bids and as such have a value of zero. The other arcs are bid arcs and have a weight corresponding to the value of the bid.

to $\langle 0, 4\rangle$, from $\langle 1, 3\rangle$ to $\langle 1, 4\rangle$, from $\langle 2, 3\rangle$ to $\langle 2, 4\rangle$ and from $\langle 3, 3\rangle$ to $\langle 3, 4\rangle$.

The longest path in the resulting graph from node $\langle 0, 0\rangle$ to $\langle 3, 4\rangle$ has a value of 30 and goes along the following arcs. First it goes from $\langle 0, 0\rangle$ to $\langle 1, 1\rangle$, which means bid 1 is selected. Next it goes from $\langle 1, 1\rangle$ to $\langle 2, 3\rangle$, meaning bid 2 is selected. Then there are two alternative but equally good paths. The first alternative is to go from $\langle 2, 3\rangle$ to $\langle 2, 4\rangle$, thus accepting bid 5 and then go from $\langle 2, 4\rangle$ to $\langle 3, 4\rangle$ by using the zero arc. The other alternative is to go from $\langle 2, 3\rangle$ to $\langle 3, 3\rangle$ by using the zero arc and then going from $\langle 3, 3\rangle$ to $\langle 3, 4\rangle$, thus accepting bid 5. The optimal solution of the winner determination problem in this instance is thus to accept bids 1, 2 and 5. The corresponding value is 30.

Proof of correctness

Theorem 2.1. *The WDP with n connected and gap-free bids on m items that are arranged in k rows, can be solved by solving a shortest path problem in an acyclic graph $G = (V, A)$ with $O(m^k)$ nodes and $O(m^{k-1}(n + m))$ arcs for each $k \geq 1$.*

Proof. We show that the dynamic program for k rows, as described in 2.3.1, solves the winner determination problem to optimality, i.e. a longest path in G corresponds to a solution to the winner determination problem with the same value, and vice versa. We prove this by establishing a one-to-one correspondence between a feasible set of bids and a path in the graph G from $\mathbf{0} = \langle 0, \ldots, 0\rangle$ to $\boldsymbol{m} = \langle m_1, \ldots, m_k\rangle$.

First, we show how a given set of non-overlapping bids corresponds to a path in G. We order the given bids in a sequence such that a bid containing item $x \in X_j$ comes before a bid containing item $y \in X_j$ when $x \prec y$ (for each $j \in \{1, \ldots, k\}$). Notice that the bids being gap-free and connected implies that at least one such a sequence exists. The path in G corresponding to this sequence of bids consists of a single bid arc for each bid in the sequence, and zero arcs in between the bid arcs. Let us assume a partial path in

G starting at $\mathbf{0}$ going to node $\boldsymbol{y} = \langle y_1, \ldots, y_k \rangle$ has been found that corresponds to the first u bids in the sequence. Thus, the first u bids have allocated items up to y_1 in row 1, ..., and up to y_k in row k. We show how to extend this partial path to incorporate the $(u+1)$-th bid.

Let b be the $(u+1)$-th bid. By definition $x^b_{s(b)+1}, \ldots, x^b_{f(b)-1}$ are the leftmost items in the rows $s(b)+1, \ldots, f(b)-1$ where bid b is present. By construction of the sequence we have that $y_i \prec x^b_i$ for $i = s(b)+1, \ldots, f(b)-1$. Thus, we can use zero arcs starting in $\boldsymbol{y}$ to bring us to node $\boldsymbol{y'} = \langle y_1, y_2, \ldots, y_{s(b)}, x^b_{s(b)+1} - 1, \ldots, x^b_{f(b)-1} - 1, y_{f(b)}, \ldots, y_k \rangle$. Now we select the bid arc corresponding to bid b that leaves node $\boldsymbol{y'}$.

Second, given a path in G from $\mathbf{0}$ to $\boldsymbol{m}$, it is obvious how a feasible set of bids is chosen: simply take the bids corresponding to the bid arcs in the path. There can be no overlap between any pair of these bids, since there are no arcs in G from nodes $\boldsymbol{x}$ to $\boldsymbol{y}$ for which any of the components of $\boldsymbol{x}$ succeeds a component of $\boldsymbol{y}$. The value of the set of bids coincides with the length of the path.

Finally, it is not difficult to verify that the graph G is acyclic, and hence a longest path can be found efficiently by solving a shortest path problem in G with edge weights multiplied by -1. □

2.3.2 A generalization of the k row dynamic program

In Section 2.3.1, we described a dynamic programming algorithm that works in a geometrical setting corresponding to k rows. Now consider a situation more general than the geometric setting with k consecutive rows. Let us assume we are given k rows that display arbitrary adjacencies. We model this using a so-called *layout graph*, which has a node for every row, and two nodes are connected if the corresponding rows are incident to each other. Clearly, the setting of the previous section boils down to the layout graph being a path on k nodes, an example of which is given in Figure 2.8.

In this section, we allow any simple graph, i.e. an unweighted and undirected graph containing no graph loops or multiple edges,

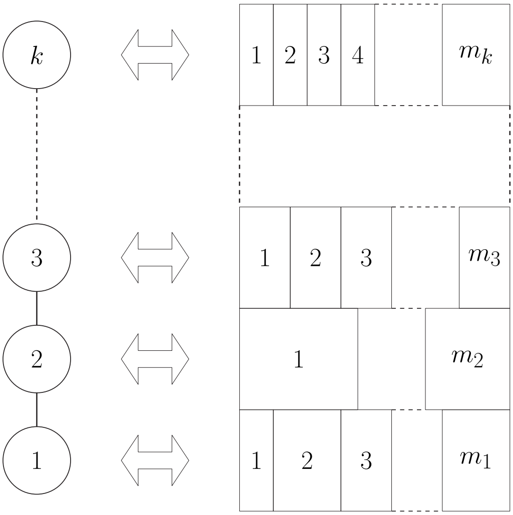


Figure 2.8: Graph layout of a standard k-row problem.

on k nodes to be the layout graph. We give an example of a general 6-row problem in Figure 2.9. In that figure, the row corresponding to node 2 is adjacent to the rows corresponding to nodes 1, 3, 4, 5, and 6. We also remark that this no longer corresponds to the rows being in a two-dimensional plane. However, there can still be practical application of generalized graph layouts. Consider for example the situation depicted in Figure 2.10. That graph layout corresponds to four rows of items that form a cylindrical shape, which could e.g. represent an advertising column.

An item $i \in \cup_j X_j$ is characterized by two things.

1. The row it belongs to, called $r(i)$, with $r(i) \in \{1, \ldots, k\}$.

2. The interval $[s_i, f_i]$ it occupies.

Given a layout graph $G = (V, E)$, we now generalize Definition 2.1.

Definition 2.4. *A pair of items i and j are adjacent if and only if either $r(i) = r(j)$ and $[s_i, f_i] \cap [s_i, f_i] \neq \varnothing$, or $(r(i), r(j)) \in E$ and $[s_i, f_i] \cap [s_i, f_i] \neq \varnothing$.*

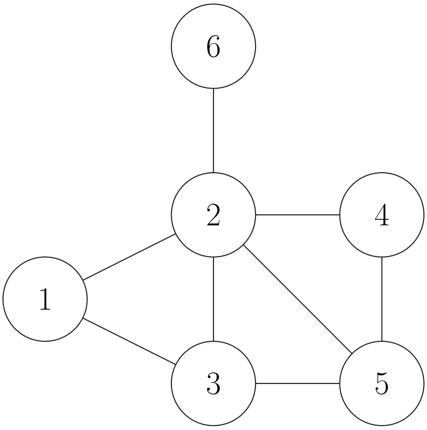


Figure 2.9: Graph layout of a general 6-row problem.

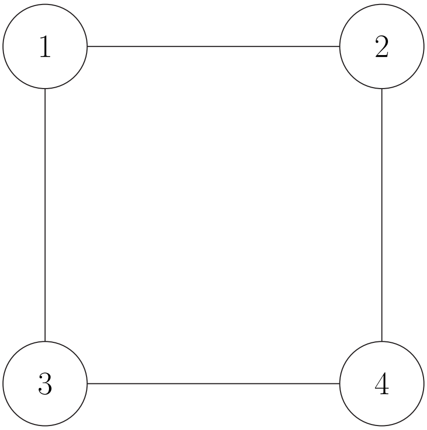


Figure 2.10: Graph layout of a 4-row problem.

With this definition of adjacency, we observe that the dynamic program still works for any arbitrary layout graph. Indeed, consider two disjoint bids A and B, each seen as a set of items. If there is no row that contains items from A and items from B, we say that bids A and B are not *comparable*. If there is a row containing items from both A and B, gap-freeness ensures that all items from A are to the left (or to the right) from the items of B. Assume that on this row $A < B$. We claim that connectedness implies that on each row where items of A as well as items of B are present, we must have $A < B$, meaning that A and B are comparable. Consequently, a set of disjoint bids forms a partial order, implying the existence of a sequence of these bids.

2.3.3 The complexity of winner determination for bids in a grid

In this section we assume that a $k \times q$ grid is given, with $m = k \times q$ items (for instance representing seats in a grandstand, or a theater), and that connected bids are given. Naturally, the dynamic program for k rows, as presented in Section 2.3.1, can also be used to solve instances of a grid setting, as this is a special case of our geometric setting described in Section 2.2. However, if we consider instances for the grid setting with any number of rows (i.e. we consider k as part of the input), the dynamic program of Section 2.3.1 can no longer guarantee a polynomial running time as the graph consists of $\mathrm{O}(m^k)$ nodes and $O(m^{k-1}(n+m))$ arcs. As mentioned in Section 2.1, Rothkopf et al. (1998) find that if bids are allowed only on singletons, full rows, and full columns, the problem is easy to solve.

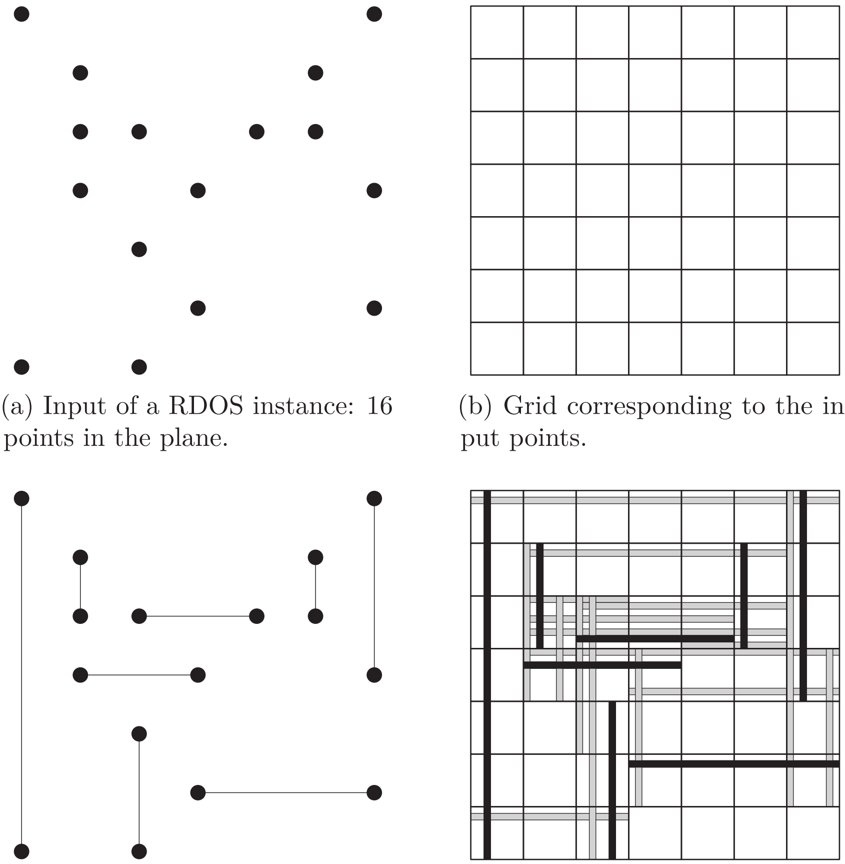
Probably the simplest bids that use multiple rows and columns are 2×2 bids (a 2×2 bid is a bid on cells $(i, j), (i, j+1), (i+1, j), (i+1, j+1)$) for some row i and column j. However, when each bid is a 2×2 bid, the complexity of the winner determination problem follows directly from the *tile salvage problem*. In this problem, an $k \times k$ grid is given, together with a set of unit squares that have been removed from this grid. The tile salvage problem is to find the maximum number of non-overlapping $x \times y$ tiled rectangles. Berman et al. (1990) show that the tile salvage problem is NP-complete, even for 2×2 tiles. Hence, our problem is hard even if only bids on 2×2 rectangles are allowed.

For connected bids in a $k \times q$ grid, the only setting whose complexity is open is a setting where each bid is either a *row* bid or a *column* bid. We say that a bid is a row bid (column bid) when it consists of consecutive items on some single row (column). Note that in this setting bids need not be on an entire row/column, but can be on a part of a row or a column as well. Obviously, if, in a grid of size $k \times q$, all bids are row bids (or all bids are column bids), the problem decomposes into k (q) independent single row (column) problems; however, if the instance contains both row bids and column bids, the complexity follows from the following observation.

Theorem 2.2. *The winner determination problem in a grid where each bid is a row bid or a column bid, is NP-hard.*

Proof. The following question is known to be NP-complete (Rendl and Woeginger, 1993). Given $2n$ distinct points in the plane, do there exist n axis-aligned, non-overlapping line segments each connecting a pair of points such that each point is connected to exactly one other point? A segment is called *axis-aligned* when the two points it connects either share an x-coordinate, or share a y-coordinate. Like Rendl and Woeginger (1993), we will call this problem RDOS (reconstruction of sets of disjoint orthogonal segments).

Let us now build an instance of the winner determination problem in a grid. For each distinct y-coordinate in the instance of RDOS there is a row in our problem, and for each distinct x-coordinate there is a column in our problem. This specifies the grid. Every cell of the grid corresponds to an item. An example can be seen in Figure 2.11a and 2.11b. For each pair of points sharing a y-coordinate (x-coordinate), there is a row (column) bid with value 1, containing all items in between the two points sharing the y-coordinate (x-coordinate). This completely specifies an instance



(a) Input of a RDOS instance: 16 points in the plane.

(b) Grid corresponding to the input points.

(c) A solution connecting all 16 points with 8 non-overlapping line segments.

(d) Black rectangles correspond to bids in an optimal solution, grey rectangles are other bids.

Figure 2.11: Transformation of a RDOS instance to an instance of the winner determination problem in a grid with row and column bids.

of the winner determination problem in the grid. An example can be seen in Figure 2.11c and 2.11d. Now, if total revenue of the corresponding auction has a value of n, then apparently there are n row and column bids that do not overlap. These n bids correspond to n axis-aligned segments, and the answer to the question is yes. Finally, if the answer to the question is yes, there exist n non-overlapping row and column bids. □

Notice that there is an easy 2-approximation algorithm for this setting. The approximation goes as follows. First, consider only row bids and solve the corresponding winner determination problem by solving the problem for each row. Next, perform a similar procedure for the sets of column bids. Finally, we take the best result of these two feasible solutions. It is easy to see that this is in fact a 2-approximation. Recall that solving the winner determination problem for connected bids on a single row is polynomially solvable.

2.4 Variants

In this section, we take a more detailed look at the case of two rows, showing the impact of each of the two assumptions (connected and gap-free) on the computational complexity of the WDP. Finally, we show how the dynamic program can be generalized to treat the case of three rows and connected bids.

2.4.1 The case of two rows and gap-free bids

We relax here the condition of connectedness; we only assume that bids are gap-free (we have two rows however). We claim that, in this case, the WDP becomes a special case of the problem of finding a maximum-weight independent set in a graph that is the edge-union of two interval graphs. Indeed, observe that since a bid is gap-free we can see each bid as the union of a set of consecutive items in row 1 and a set of consecutive items in row 2. By concatenating row 1 and row 2 into a single row, one can view each each bid as consisting of two intervals, a left and a right interval. The resulting intersection graph has a node for each bid, and two nodes are connected if either their left intervals, or their right intervals (or both) overlap; in other words, the resulting intersection graph is a 2-union graph. It is shown in Bar-Yehuda et al. (2006), that the maximum-weight independent set problem is NP-hard on 2-union graphs, see also van Bevern et al. (2015). Note however that the intersection graph resulting from the 2-row problem we investigate is a special case of 2-union graphs. Indeed, in our special case all left intervals are to the left of all right intervals, which is not necessarily the case in a 2-union graph. However in the context of computational biology this precise special case has been studied by Vialette (2004).

Lemma 2.1. *The WDP with n gap-free bids on m items that are arranged in two rows, is NP-hard.*

Proof. See the proof of Proposition 7 in Vialette (2004). □

2.4.2 The case of two rows and connected bids

Consider the case where bids are still connected, but not necessarily gap-free. Figure 2.12 shows an example of a bid that has 2 gaps. Given a bid b, the set of items that are in gap(s) of this bid b is given by $G(b) = \{x \notin b : \exists u, v \in b \text{ with } u \prec x \prec v\}$. In case $G(b)$ is empty, b is gap-free; otherwise $G(b)$ consists of, say $p(b)$ ($p(b) < m$), connected itemsets, each representing a single gap. More precisely,

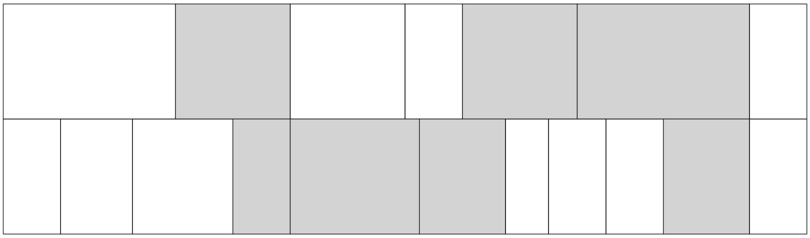


Figure 2.12: A bid with 2 gaps in a 2-row problem.

let $H(G(b))$ be the item graph corresponding to the items in $G(b)$; each of the $p(b)$ connected components of $H(G(b))$ corresponds to items making up a single gap. We write $G(b) = \bigcup_{\ell=1}^{p(b)} G^{\ell}(b)$, where $G^{\ell}(b)$ represents the items present in the ℓ-th gap of bid b where $1 \leq \ell \leq p(b)$.

Theorem 2.3. *The WDP with n connected bids on m items that are arranged in two rows, can be solved in polynomial time.*

Proof. Observe that, for each $\ell = 1, \ldots, p(b)$ and $b \in B$, the itemset $G^{\ell}(b)$ consists of items on a single row (otherwise b would be disconnected). Let us now consider an instance defined by itemset $G^{\ell}(b)$, and by all bids $b' \in B$ that are contained in this itemset. Since each b' is connected (by assumption) and since $G^{\ell}(b)$ consists of items on a single row (see above), we can easily compute the value of this instance (denoted by $v(G^{\ell}(b))$) by using Theorem 2.1 with $k = 1$. Given a bid b, we do this for each $\ell = 1, \ldots, p(b)$ finding the values $v(G^{\ell}(b))$ by applying Theorem 2.1 for $k = 1$ $O(m)$ times.

Finally, given an instance, we build a new instance where we replace each bid b that is not gap-free by a combined bid on the itemset $b \cup G(b)$, with a value $v(b) + \sum_{\ell=1}^{p} v(G^{\ell}(b))$. The resulting instance is created in polynomial time, is gap-free, and thus we can use Theorem 2.1 to solve it. □

2.4.3 The case of three rows and connected bids

Here, we show how the winner determination problem for the setting with 3 rows and connected bids can be solved as a shortest path problem, using a generalization of the approach described in Section 2.3.1 that can handle bids with open gaps. The main challenge in this case is how to deal with gaps that may be present in a bid.

Let us first define the concept of an instance graph H. The instance graph H has a node for each item $x \in (X_1 \setminus \{0\}) \cup (X_2 \setminus \{0\}) \cup (X_3 \setminus \{0\})$. Two nodes corresponding to items that are adjacent are connected; moreover, there is a node s in the graph which is connected to the first item in each of the three rows, and there is a node t connected to the last item in each of the three rows (see Figure 2.13 for an example).

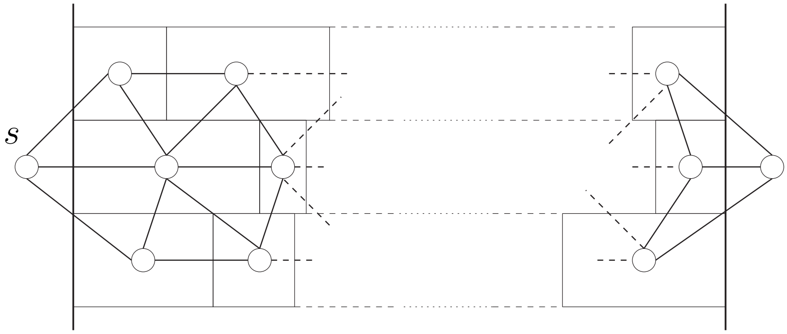


Figure 2.13: The instance graph H.

Types of gaps and bids

We distinguish two kind of gaps. To that end, consider the instance graph H, and a connected bid b, and suppose that bid b is not gap-free. Thus, each of the p gaps in bid b is represented by itemset $G^{\ell}(b)$, $\ell = 1, \ldots, p$.

Definition 2.5. *We call a gap $G^{\ell}(b)$ an* open gap *if, in the graph $H \setminus H(b)$, there is a path from each $x \in G^{\ell}(b)$ to either node s or node t. Each gap that is not an open gap is called a* closed gap.

Because there are only 3 rows, a closed gap contains items in at most 2 rows (since a closed gap on 3 rows corresponds to a bid that is not connected). Also notice that an open gap only has items in the second row.

In Figure 2.14 there are 3 examples. In the top example, there is one gap with items on the first and second row. However, since there is no path in $H \setminus H(b)$ from any of the $x \in G^1(b)$ to either s or t, it is a closed gap. In the middle example, there is a gap with items on the second row. There exists a path in $H \setminus H(b)$ from each $x \in G^1(b)$ to s. Therefore, the gap in this bid is an open gap.

In the bottom example of Figure 2.14 there are 4 gaps. The first gap, $G^1(b)$ on the left has one item on the first row; the second gap, $G^2(b)$, has one item on the third row; the third gap, $G^3(b)$, has one item on the first row and one item on the second row. There is no path in $H \setminus H(b)$ from any of the $x \in G^\ell(b)$ to either s or t for $\ell = 1, 2, 3$. This means that these three gaps are closed gaps. Finally, the fourth gap $G^4(b)$, is on the right and has one item on the second row. There is a path in $H \setminus H(b)$ from the item in $G^4(b)$ to t, making this gap an open gap.

We now partition the class of connected bids in two disjoint subclasses according to the following definition.

Definition 2.6. *If a bid b has at least one open gap, it belongs to the subclass called* open bids*. The set of open bids is $\mathcal{B}_{open}$. If b has no open gaps, it belongs to the subclass of* closed bids*. The set of closed bids is $\mathcal{B}_{closed}$.*

Consider a bid and its closed gaps. For each such closed gap, we solve the corresponding instance, yielding a value v. We then replace the bid with a combined bid that has the closed gaps filled and its original value increased by v. From Theorem 2.3 it follows that this operation can be done in polynomial time (recall that a closed gap contains items on at most two rows).

After this preprocessing of the bids, all closed gaps in all bids are ‘filled’ optimally. See for example Figures 2.15, 2.16 and 2.17. In Figure 2.15 there is a gap spanning the top and middle row, which is then filled optimally by solving a subproblem. In this case, the entire gap has been covered by other bids, but this is not necessarily always true. In Figure 2.16, there is a gap only on the middle row. It is filled optimally by solving a subproblem which in this case only covers half the space in the gap. In Figure 2.17, there is both a closed gap and an open gap. The closed gap has been filled optimally by solving a subproblem.

After this preprocessing step we can ignore the gaps in closed bids, because they are all filled optimally (replacing the closed bid with a combined bid). In an open bid, all closed gaps have been filled

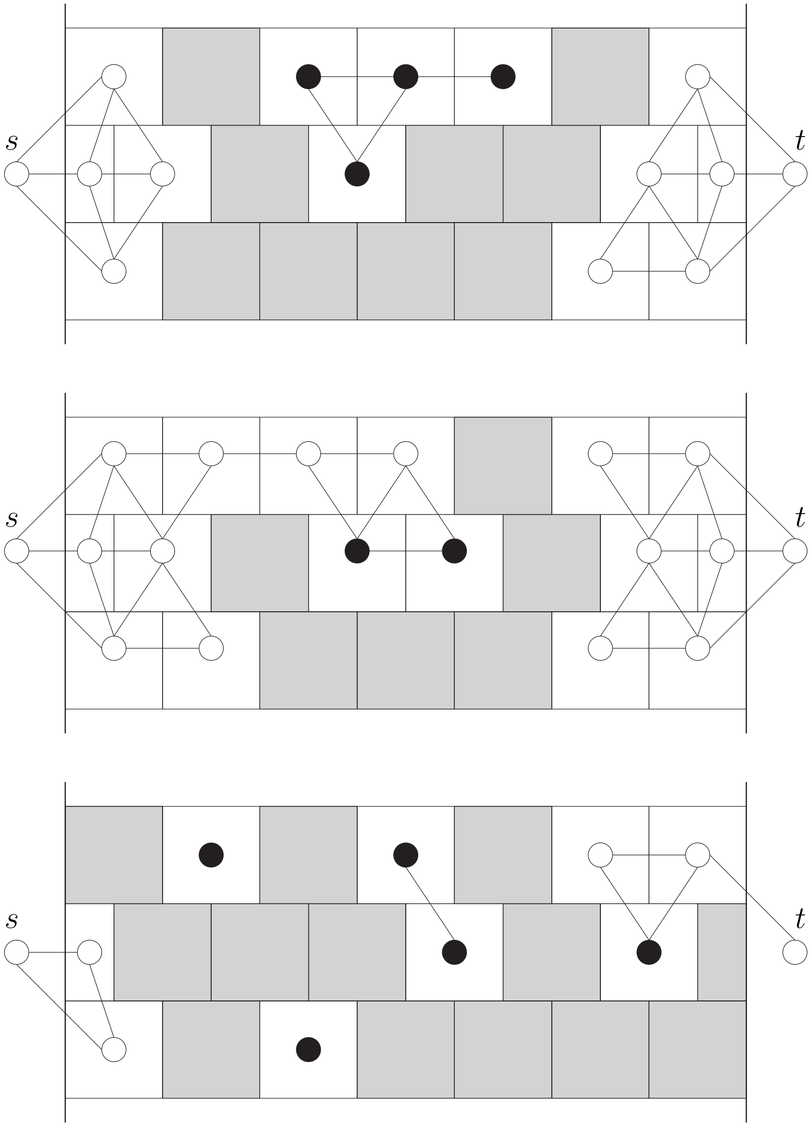


Figure 2.14: Examples of graphs $H \setminus H(b)$. Black nodes correspond to items in a gap.

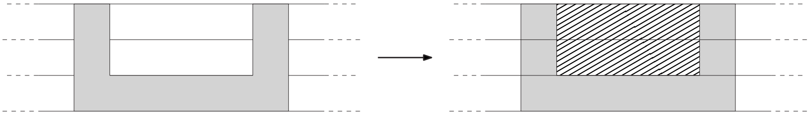


Figure 2.15: A closed gap spanning two rows.

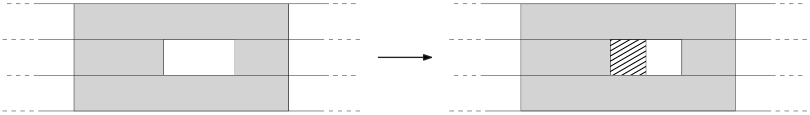


Figure 2.16: A closed gap spanning one row.

as well, however, there is always at least one open gap remaining. Note that of course $\mathcal{B}_{\text{open}} \cup \mathcal{B}_{\text{closed}} = \mathcal{B}$ and $\mathcal{B}_{\text{open}} \cap \mathcal{B}_{\text{closed}} = \varnothing$ hold.

A polynomial-time algorithm

We show how the winner determination problem for the setting with 3 rows and connected bids can be solved as a shortest path problem. We construct the graph $G = (V, A)$ as follows.

The nodes We define $\langle x_1, x_2, x_3, b\rangle$ (quadruples), where $x_1 \in X_1$, $x_2 \in X_2$, $x_3 \in X_3$, $b \in (\mathcal{B}_{\text{open}} \cup \varnothing)$. Every such quadruple corresponds to a node in V where items $\{0, \ldots, x_1\} \cup \{0, \ldots, x_2\} \cup \{0, \ldots, x_3\}$ have been allocated. If $b = \varnothing$, we are in a *bid-independent state* (which corresponds to the states described in Section 2.3.1). If $b \neq \varnothing$, then we are in a *bid-dependent state* where we have to take into account one or more open gaps and have also assigned items $\{x'_b, \ldots, x''_b\}$ in the middle row with $x_2 \prec x'_b$. Specifically, each open bid b is characterized by x'_b and x''_b, where x'_b is the leftmost item of the last contiguous set of items on the second row included in b and x''_b is the rightmost item of the last contiguous set of items on the second row included in b. Note that it is possible that $x'_b = x''_b$.

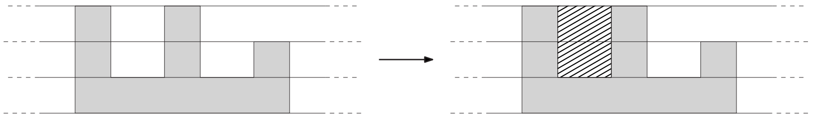


Figure 2.17: A closed gap spanning two rows and an open gap.

The arcs There are 2 types of arcs. The first type of arcs are zero arcs, which are are used to handle items not included in the set of winning bids. These arcs are not associated with any bid and thus have length 0. We distinguish 3 different types of zero arcs:

- Arcs between two bid-independent nodes:
  - from $\langle x_1, x, y, \varnothing \rangle$ to $\langle x_1 + 1, x, y, \varnothing \rangle$, $\forall x_1 \in X_1 \setminus \{m_1\}$, $x \in X_2$, $y \in X_3$,
  - from $\langle x, x_2, y, \varnothing \rangle$ to $\langle x, x_2 + 1, y, \varnothing \rangle$, $\forall x \in X_1$, $x_2 \in X_2 \setminus \{m_2\}$, $y \in X_3$, and
  - from $\langle x, y, x_3, \varnothing \rangle$ to $\langle x, y, x_3 + 1, \varnothing \rangle$, $\forall x \in X_1$, $y \in X_2$, $x_3 \in X_3 \setminus \{m_3\}$.
- Arcs between two bid-dependent nodes: from $\langle x, x_2 - 1, y, b \rangle$ to $\langle x, x_2, y, b \rangle$, $\forall x \in X_1$, $x_2 \in X_2 \setminus \{0\} : x_2 \prec x'_b$, $y \in X_3$, $b \in \mathcal{B}_{\text{open}}$.
- Arcs between bid-dependent and bid-independent nodes: from $\langle x, x'_b - 1, y, b \rangle$ to $\langle x, x''_b, y, \varnothing \rangle$, $\forall x \in X_1$, $y \in X_3$, $b \in \mathcal{B}_{\text{open}}$.

The second type of arcs are those which are associated with actual bids. The lengths of these are equal to the value of the corresponding (combined) bid. Note that there may be multiple arcs corresponding to the same bid. We now describe the 4 components of a node $\langle x_1, x_2, x_3, b' \rangle$ that make up a starting node for an arc that corresponds to a connected bid $b \neq b'$:

- First tuple component x_1: if $b \cap X_1 \neq \varnothing$, then $x_1 = x - 1$, where x is the leftmost item in row 1 included in b. Otherwise all elements in X_1 are possible values for x_1, i.e. multiple arcs will need to be constructed.

- Second tuple component x_2: if $b \cap X_2 \neq \varnothing$, then $x_2 = x - 1$, where x is the leftmost item in row 2 included in b. Otherwise all elements in X_2 are possible values for x_2, i.e. multiple arcs will need to be constructed.

- Third tuple component x_3: if $b \cap X_3 \neq \varnothing$, then $x_3 = x - 1$, where x is the leftmost item in row 3 included in b. Otherwise all elements in X_3 are possible values for x_3, i.e. multiple arcs will need to be constructed.

- Fourth tuple component b': $b' = \varnothing$ or $b' \in \mathcal{B}_{\text{open}}$ for which the following holds:

  - $b \cap b' = \varnothing$ (no overlap) and
  - $\exists x \in b \cap X_2, \exists x'$ and $x'' \in b' \cap X_2 : x' \prec x \prec x''$ (b has an item in an open gap of b') and
  - $\nexists x \in b \cap X_2, x' \in b' \cap X_2 : x \prec x'$ (b does not have an item to the left of b' in the second row)

Now that we have determined all possible starting nodes for every connected bid, we have to determine the end nodes. End nodes for arcs depend on two things: the starting node and whether the bid b to which the arc corresponds is a closed or an open bid. We distinguish four cases.

- Case 1: bid-independent starting node, closed bid.

- Case 2: bid-independent starting node, open bid.

- Case 3: bid-dependent starting node, closed bid.

- Case 4: bid-dependent starting node, open bid.

We will now discuss how the end node is constructed from the starting node in each case.

- Case 1: for every row for which there is an item $x \in b$, change the corresponding tuple component to the rightmost item included in b in that row, otherwise keep the value of the starting node. The fourth tuple component remains the same as the starting node, which is $\varnothing$.

- Case 2: for the first and third row for which there is an item $x \in b$, change the corresponding tuple component to the rightmost item of that row included in b, otherwise keep the value of the starting node. For the second row: find the leftmost contiguous interval included in b and change the second tuple component to the rightmost item in that interval. The fourth tuple component will change to b.

- Case 3: for every row for which there is an item $x \in b$, change the corresponding tuple component to the rightmost item included in b in that row, otherwise keep the value of the starting node. The fourth tuple component remains the same as the starting node.

- Case 4: let b' be an open bid with its leftmost item in row 2 to the left of leftmost item of open bid b in row 2. Observe that the itemset $b' \cup b$ may contain a closed gap: indeed there are two basic cases depending on whether the rightmost item in row 2 in b precedes (Figure 2.18) or succeeds (Figure 2.19) the rightmost item in row 2 in b'. In both cases, the value of the arc will be increased with the optimal value of a subproblem on the second row limited to the shaded area. In other words, we construct a combined bid. In the case of Figure 2.18, the first three tuple components are changed according to the rightmost item included in b in that row, otherwise keeping the value of the starting node. The fourth tuple component will remain the same. In the example, the arc would go from $\langle x_1 - 1, x_2 - 1, y_3, b' \rangle$ to $\langle x_1, x'_2, y_3, b' \rangle$. In the case of Figure 2.19, the first three tuple components are changed according to the rightmost item included in b in that row, otherwise keep the value of the starting node. However, the fourth tuple component will be changed to b. In the example, the arc would go from $\langle x_1 - 1, x_2 - 1, y_3, b'\rangle$ to $\langle x_1, y_2', y_3, b\rangle$.

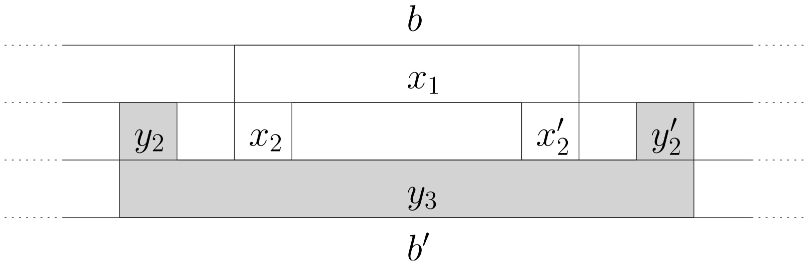


Figure 2.18: Two open bids creating an extra gap: option 1.

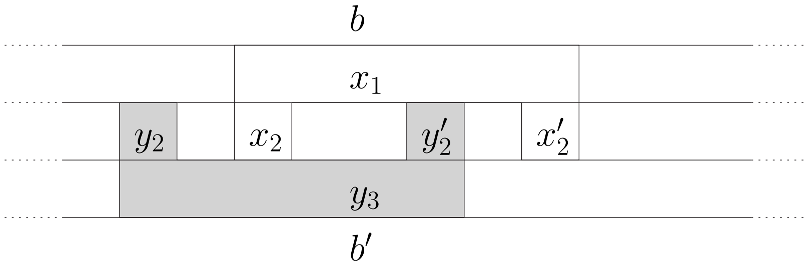


Figure 2.19: Two open bids creating an extra gap: option 2.

Shortest path We now compute a longest path from node $\langle 0, 0, 0, \varnothing\rangle$ to node $\langle m_1, m_2, m_3, \varnothing\rangle$. The length of this path corresponds to the optimal revenue of the auction, and the winning bids can be derived from the arcs in the path. Notice that $G = (V, A)$ is acyclic by construction and consists of $O(nm^3)$ nodes and $O(n^2m^3)$ arcs. Hence, a longest path can be found efficiently by solving a shortest path problem in $G = (V, A)$ with edge weights multiplied by -1. In the next section, we prove the correctness of this algorithm.

Proof of correctness

In order to prove the correctness of the algorithm described in Section 2.4.3, we show that (1) each path from node $\langle 0, 0, 0, \varnothing \rangle$ going to node $\langle m_1, m_2, m_3, \varnothing \rangle$ corresponds to a feasible allocation for the auction, and (2) vice-versa. Recall that after the preprocessing step described in Section 2.4.3, all closed gaps have been filled, resulting in combined bids and corresponding arcs in G. As the individual bids corresponding to these arcs can easily be traced, we will ignore closed gaps in the remainder of this proof.

(1) Intuitively, consider a path starting from node $\langle 0, 0, 0, \varnothing \rangle$ and going to node $\langle m_1, m_2, m_3, \varnothing \rangle$. This path corresponds to an allocation for the auction by accepting the bids corresponding to the arcs associated with bids (the zero arcs can obviously be ignored). In order to show that this allocation is feasible, we need to argue that no pair of bids has overlap. Notice that for all arcs in the graph, when comparing the end node with the start node, none of the first 3 tuple components decreases, and at least one increases. This means that each arc represents moving to the right on at least one row, and that moving (back) to the left is not possible. For arcs with a bid-independent starting node, the start node corresponds to the leftmost items on each row included in the bid. Hence, overlap between the bid corresponding to this arc and any of the bids corresponding to previous arcs in the path is not possible. For arcs with bid-dependent starting nodes, more care is needed. Consider an arc whose starting node has b' as the fourth tuple component. By construction, for each such arc corresponding to a bid b, there is no overlap between b' and b. Furthermore, as the endpoint of this arc determines to what extent the open gap(s) of b' has been filled by b, we also avoid overlap between b and the corresponding bid of a possible next arc with value b' in the fourth tuple component of its starting node. The only way to move from a bid-dependent to a bid-independent node is through a zero arc, which ensures that the value for the second tuple corresponds with the rightmost item

on the second row of the open bid b', thereby excluding overlap between this bid and bids corresponding to subsequent arcs.

(2) Consider a feasible solution for the winner determination problem (i.e. no pair of bids in the allocation overlaps). By construction, for each (combined) bid in the allocation at least one arc in the graph exists. We show how to identify a path in G that corresponds with the bids in the allocation. We order the bids in a sequence such that bid p comes before bid q if p contains an item $x \in X_2$ for which $x \prec y$ for all $y \in X_2$ contained in bid q. In other words, we order the bids based on their leftmost item on the second row. Since we assume that bids are connected, bids that do not contain items on the second row have all items either on the first row or on the third. These bids should be inserted in the order such that a bid containing item $x \in X_j$ comes before a bid containing item $y \in X_j$ when $x \prec y$ (for each $j \in \{1, 3\}$). Recall that these bids are not used to fill closed gaps, as we handled this in the preprocessing step. Hence, at least one such sequence exists.

The path in G corresponding to this sequence of bids has a single bid arc for each (combined) bid in the sequence, and zero arcs in between the bid arcs. Let us assume a partial path in G starting at $\langle 0, 0, 0, \varnothing \rangle$ going to node $\langle u_1, v_1, w_1, b' \rangle$ has been found that corresponds to the first k bids in the sequence. We show how to extend this partial path to incorporate the $(k+1)$-th bid, say bid b. We discern 3 situations:

- $b' = \varnothing$: if b has items on each row, we select the corresponding arc starting at node $\langle u_2 - 1, v_2 - 1, w_2 - 1, \varnothing \rangle$, where u_2, v_2, and w_2 are the leftmost items in b. If b has no items on one or more rows, we select the arc starting from the node with tuple value u_1, v_1 and/or w_1 for the respective row(s) on which b has no items.

- $b' \neq \varnothing$ and b is a closed bid: we select the corresponding arc starting at node $\langle x, y, z, b' \rangle$, where $x = u_1$ ($y = v_1$,$z = w_1$) if b does not include an item on the first (second, third) row, and

$x = u_2 - 1$ ($y = v_2 - 1$,$z = w_2 - 1$) otherwise (where u_2, v_2, and w_2 are the leftmost items in b).

- $b' \neq \varnothing$ and b is an open bid: we select the same arc as in the previous case, however, in this case one or more bids on items in row 2 may be enclosed between bids b and b' (see Figures 2.18 and 2.19). As the value of these bids in included in the weight of the arc corresponding to b, we can remove these bids from the sequence (observe that these bids succeed b).

Notice that from all bid arcs that correspond to this bid b, we select one, and that we can always reach the selected arc from $\langle u_1, v_1, w_1, b' \rangle$ using zero arcs. Next, we iteratively select the next bid in the order, and proceed analogously. After having treated the last bid in the order, if the end node of the corresponding arc is not $\langle m_1, m_2, m_3, \varnothing \rangle$, we connect to this node using zero arcs. The following result is now apparent.

Theorem 2.4. *The WDP with n connected bids on m items that are arranged in three rows can be solved by solving a shortest path problem in a graph with $O(nm^3)$ nodes and $O(n^2m^3)$ arcs.*

2.5 Conclusion

We study the winner determination problem for a combinatorial auction with a specific geometric structure. We argue that this structure is relevant, as it occurs in real estate, plots of land, mineral rights, and theaters and stadium seats. The complementarities present in these situations offer great potential for combinatorial auctions. We point out that the items need not be rectangular but can be of any shape. In fact, if the itemset can be partitioned into k ordered subsets (rows), such that the adjacency relations between pairs of items on consecutive rows are consistent with the ordering of the items in each row, our framework applies.

With our dynamic programming algorithm, we present auctioneers a tool that enables them, under some reasonable assumptions on the bids and with a fixed number of rows, to efficiently compute the winning bids. Next, we complement existing results by showing that bidding in a grid is difficult, even when only row and column bids are allowed, if the number of rows is part of the input. We further investigate the precise impact of our assumptions.

Solving the winner determination problem efficiently is an essential component of mechanism design. As this chapter assumes the bids are given, future research that focusses on determining accompanying auction rules, and studies their impact on bidding strategy, efficiency and revenue would be valuable. Finally, our results may also prove useful for experimental research: our dynamic program will allow researchers to study bidder behavior in larger settings, involving more items and bidders than considered so far.

Chapter 3

Threshold and Coordination Problems in Combinatorial Auctions

> Any coalition has its troubles, as every married man knows.
>
> *Arthur Hays Sulzberger*

3.1 Introduction

Combinatorial auctions (CAs) are allocation mechanisms that enable selling and buying multiple (indivisible) items simultaneously. In fact, CAs allow bidders to bid on packages of items and the auctioneer can allocate any package only in its entirety to the corresponding bidder. CAs have established themselves as a viable allocation mechanism in settings where (i) market prices are not readily available (otherwise there is no need for an auction), and

This chapter is joint work with Prof. Dr. Dries R. Goossens and Prof. Dr. Frits C.R. Spieksma.

(ii) bidders have sub- or super-additive valuations (otherwise there is no need for a combinatorial auction, since sequential single item auctions would suffice). Combinatorial auctions offer the possibility for a *coalition* of bids on small packages to jointly outbid a single bidder's claim on the complete set of items. However, two hurdles need to be overcome before a coalition can become winning.

(i) The ***coordination problem***. As each item can be allocated at most once, bidders need to coordinate their bids and bid on complementary (i.e. non-overlapping) sets of items. The coordination challenge lies in bidders having to discover such a set of individually profitable and collectively complementary packages, given that the number of possible packages rises exponentially with the number of items. This is complicated by the existence of cognitive limits on the number of packages people can concentrate on during the auction. For instance, experimental research by Scheffel et al. (2012) has shown that bidders only bid on six to ten different packages, independent of the auction format, although they had a multitude of packages with positive valuations to choose from. Kagel et al. (2010) also find that bidders only bid on a small number of packages, and that the majority of placed bids are on the myopically profitable packages. Furthermore, coordination is hindered by the assumption that a bidder only knows his/her private valuation for these packages, and not the preferences of other bidders. In fact, in order to mitigate collusion, it makes sense to restrict communication between bidders (see e.g. Cramton and Schwartz (2000)).

(ii) The ***threshold problem***. Even if the coordination problem is overcome and a set of disjoint packages for which the combined valuation exceeds the currently winning bid is somehow identified, the task of determining appropriate bid prices to displace the currently winning bid still remains. A complicating factor is that each bidder in a coalition has an interest not to increase his/her bid. Indeed, the forgone revenue from unilaterally increasing one's bid falls entirely on the cooperating bidder while the benefits extend to the non-cooperating bidders as well. Note that, as Bykowsky et al. (2000) point out, the threshold problem is strictly speaking not a free-rider problem. In a true free-rider problem, the dominant strategy is never to cooperate. However, in the context of the threshold problem, coalition members may still have some incentive to contribute to the effort to overcome the threshold, since being the only bidder to cooperate will typically still be preferable to not cooperating and winning nothing.

Both the coordination problem and the threshold problem are solved in the well-known Vickrey-Clarke-Groves (VCG) auction. Indeed, in the VCG auction, it is a dominant strategy for bidders to report their valuations truthfully, which takes care of the threshold problem, and hence to bid on all packages for which they have a positive valuation, which deals with the coordination problem. However, the VCG auction is rarely used in practice due to a number of issues, such as very low auction revenues (see Ausubel and Milgrom (2006)). Furthermore, there is still the issue of cognitive limits, making it unrealistic that bidders would effectively bid on each package they value.

In the literature on combinatorial auctions, as far as we are aware, the term "threshold problem" was coined by Rothkopf et al. (1998), although the problem itself was mentioned already in e.g. Banks et al. (1989). Rothkopf et al. (1998) explain the problem by means of an example with two items, where two bidders need to increase their bids on individual items in order to outbid a third bidder's package bid. The coordination problem does not play a role in this example. Bichler et al. (2017) interpret the coordination problem in a very similar way as we do, and point out that this problem has largely been ignored in the game-theoretical literature on combinatorial auctions. Indeed, several authors do not seem to make a clear distinction between the coordination and threshold problem, or simply overlook the coordination problem. Day and Raghavan (2008) define the threshold problem in a broad sense as "*the potential inability of an individual bidder to overturn an inefficient bid for a larger package without the coordinated efforts of other bidders*". Brunner et al. (2010) use a similar definition: "*Consider a situation in which a large bidder submits a package bid* *for several licences. If other bidders are interested in buying different subsets of licenses contained in the package, they might find it hard to coordinate their actions, even if the sum of their values is higher than the value of the package to the large bidder (the threshold problem)*". Further along this line, other authors use "threshold problem" for both the coordination and the threshold problem. Scheffel et al. (2012) write: "*When some small bidders have to coordinate their bids to outbid a bidder interested in a package covering many items, they are confronted with the threshold problem*". Chernomaz and Levin (2012) also seem to mix the coordination and threshold problem: "*In this situation, the single-item bidders face a coordination problem. They have an incentive to free-ride as either of their bids could be the one to push their sum above the threshold necessary to top the package bid*".

A first contribution of this chapter is that it clarifies the hitherto vaguely used concepts of coordination and threshold problems. Moreover, we develop a quantitative measure to express the severity of both problems in Section 3.4. To the best of our knowledge, we are the first to do this. Banks et al. (2003), however, touched upon the topic by introducing two effects, intended to design an experimental setting that could assess whether several auction mechanisms were able to deal with the threshold problem. The gain effect measures "*the relative value of the optimal allocation, which is purposefully composed of several small bidder packages, and the next highest value allocation, which is constructed to be a single bidder's value for the large package covering an optimal set of packages*". The own effect occurs when allocating the items to a coalition of small package bids is optimal, but one of the bidders in this coalition also has a bid on a large (overlapping) package. Hence this bidder must forego his/her claim on the large package to be included in the coalition of small package bids. Clearly, as the gain effect decreases, and with the own effect present, it becomes more difficult for small bidders to outbid the large bidder.

Even though the coordination and threshold problem are also relevant in single round combinatorial auctions, this chapter focusses on iterative combinatorial auctions. An iterative auction consists of multiple rounds, such that bidders can repeatedly increase their bids and/or introduce new bids. After each round, the auctioneer can share information (i.e. feedback) with the bidders. A study on how feedback can be used to overcome the coordination and the threshold problem is the second contribution of this chapter. We propose novel types of feedback, dubbed coalitional feedback, in Section 3.5. We make use of experimental research to test the impact of different levels of feedback on auction performance. In Section 3.6, we discuss the details of the iterative combinatorial auction used in our laboratory experiments, along with the experimental design; the results are presented in Section 3.7.

3.2 Related literature

CAs are big business, having several practical applications ranging from the allocation of airport landing slots (Rassenti et al., 1982) or harbor time slots (Ignatius et al., 2014), the allocation of spectrum licenses (Jackson, 1976; McMillan, 1994; Banks et al., 2003; Plott and Salmon, 2004; Seifert and Ehrhart, 2005; Brunner et al., 2010; Scheffel et al., 2012; Bichler et al., 2013), improving the procurement of school meals (Epstein et al., 2002), the allocation of mineral/oil drilling rights (Cramton, 2007), the allocation of bus routes (Cantillon and Pesendorfer, 2006), and real estate (Goossens et al., 2014). It follows that CAs are well researched. One could argue that because every iterative CA faces coordination or threshold problems and involves some form of feedback, all papers on (experimental) CA research are relevant for this chapter. Because of the abundance of research, we were forced to make a selection. In Section 3.2.1, we discuss contributions with respect to feedback; a survey of relevant experimental research is presented in Section 3.2.2.

3.2.1 Feedback

An early form of feedback is described by Banks et al. (1989). They introduce a so-called “stand-by queue”, which allows bidders to publicly announce their willingness to pay a certain price for a specific package. Bidders can then use this information to express a bid which, combined with one or more of the bids on the stand-by queue, is able to outbid the currently winning bid. While a stand-by queue can help to overcome the coordination problem, it is less clear how it alleviates the threshold problem. Nevertheless, there is some experimental evidence suggesting that bidders were indeed able to coordinate their bids using the stand-by queue and displace large package bids when the sum of the small bidders’ valuations was higher than that of the large bidder (Bykowsky et al., 2000). A disadvantage of the stand-by queue is that the computational burden of finding a coalition that allows a bidder to become winning is placed with the bidder. Furthermore, it offers the bidders a communication and bid signaling tool that can also be used to facilitate collusion.

Adomavicius and Gupta (2005) introduce several important concepts concerning feedback: deadness and winning levels. In short, the former is the price a bidder needs to bid to have any chance of becoming winning, whereas the latter is the price that guarantees the bidder to become winning if no other bids are increased (we refer to Section 3.3.2 for a more detailed discussion). Their work serves as a foundation for bidder support systems in CAs, as it provides theoretical, algorithmic, and computational results on deadness and winning levels. Their analysis, however, is limited to OR-bidding languages, and does not hold for XOR-bidding languages. We briefly remark that XOR-bids bidding languages impose a limit of at most one winning bid per bidder, whereas this is not the case for OR-bids bidding languages (see also 3.3).

Next, Adomavicius et al. (2012) study how bidders behave in continuous CAs. Their main objective is to study how information feedback affects bidding behavior leading to differences in the retained surplus of bidders. They used baseline feedback (all bids displayed anonymously), outcome feedback (provisional winning allocation) and price feedback (deadness and winning levels). They find that price feedback leads to higher efficiencies, fewer dead bids and a higher percentage of winning bids when compared to outcome and baseline feedback. While they do not specifically look at coordination or threshold problems, this paper is nonetheless very interesting because it is the first one to employ deadness and winning levels as feedback.

Petrakis et al. (2013) build on the work by Adomavicius and Gupta (2005) and introduce deadness and winning levels for CAs that allow for XOR-bids. They define and analyze computational and game theoretical properties of deadness and winning levels. They mention the threshold problem, and the fact that often winning levels are too high for a single bidder to outbid a large bidder. As a solution they briefly suggest coalitional winning levels, which they introduce as personalized and non-linear ask prices in between deadness and winning levels. The underlying idea is that the costs to outbid the currently winning bid is shared among the bidders in a losing coalition. However, they point out that coalitional winning levels are computationally very challenging and do not change the free-rider incentive, and as such they do not further expand on this concept.

In a study related to ours, Bichler et al. (2017) introduce an ascending combinatorial auction which implements coalitional winning levels, where the cost sharing is based on the well-known Shapley value. It is important to realize an essential difference with this chapter: the coalitional winning levels in Bichler et al. (2017) are implemented as a price rule. This means that bidders either accept the suggested price, or are forced to drop out. In our laboratory experiment (see Section 3.6), we use coalitional winning levels as feedback, i.e. purely informative (bidders can bid any price they prefer). Besides numerical simulations, Bichler et al. (2017) perform lab experiments with human participants on rather small auction settings (6 items and 3 bidders). The results of their experiments indicate high economic efficiencies in ascending CAs with deadness and winning level feedback, but even higher efficiencies if the price rule based on coalitional winning levels is enforced, in addition to giving deadness and winning levels. The price rule also appears to lead to faster auctions.

3.2.2 Experimental Research

Quite some research focuses on experimentally testing CAs, either using computational experiments or by letting human/computerized bidders compete in auctions. Usually, this type of research intends to compare different auction formats, auction rules, etc. We survey those papers that deal (to some extent) with feedback and/or the coordination/feedback problem.

Banks et al. (2003) report laboratory experiments to analyze the Federal Communications Commission's (FCC) eligibility rules for spectrum auctions, and to compare the simultaneous multi-round auction (SMR) to a combinatorial multi-round auction (CMA) designed by Charles River and Associates. One of their results is that in instances where they try to create threshold problems, economic efficiency decreases notably for both the SMR and CMA auctions. However, efficiencies are higher in the CMA auctions compared to the SMR auctions, suggesting unsolved threshold problems. The feedback in the SMR auction is basic — the current highest bid on every item is given after every round. The feedback in the CMA auction is similarly basic — only the winning bids were posted for all bidders to see after every round.

Brunner et al. (2010) also report on laboratory experiments that aim to evaluate the performance of several auction mechanisms. They find that in RAD auctions (see Kwasnica et al. (2005) for more information on this format) and SMRPB auctions (SMRPB auctions are basically the FCC variant of RAD pricing, but include an XOR-bidding rule) so-called small bidders do not bid up to their values in periods in which they end up winning nothing. Coupled with far from perfect efficiency results, this is an indication of the threshold problem. The feedback in these auctions consists of the basic winning allocation feedback, along with a series of ask prices for bundles. As a side note, the SMRPB auctions were the only procedure Brunner et al. (2010) tested that had an XOR-bidding rule. XOR-bidding may require bidders to enter bids on many packages before bidders can find possible efficiency gains. While this might explain the lower efficiencies in the SMRPB auctions, this explanation is not valid for the RAD auctions.

Goeree and Holt (2010) propose a new multi-object auction design: the Hierarchical Package Bidding (HPB) auction, in which bidders can only bid on predefined packages. They compare HPB to the Modified/Flexible Package Bidding auction (MPB; basically MPB auctions are the FCC variant of RAD pricing, similar to the SMRPB auctions used in Brunner et al. (2010)). Goeree and Holt (2010) find that there were several MPB auctions in which a large bidder won many items, while this was not efficient. It appears the smaller bidders were unable to coordinate: they were bidding on packages which were overlapping, leaving unsold licenses. Efficiencies were lower in MPB auctions, which suffered from coordination problems, compared to the HPB auctions, which did not suffer from coordination problems. In the HPB auctions, the predefined packages enabled better coordination amongst bidders, with positive effects for both efficiencies and revenues, even if the predefined packages do not perfectly fit the smaller bidders' valuations. Goeree and Holt (2010) are the first to clearly demonstrate the existence of threshold problems for smaller bidders in auctions with flexible package bidding.

Kazumori (2010) compares generalized Vickrey-Clarke-Groves auctions, Simultaneous Ascending auctions and Clock-Proxy auctions in a laboratory setting. Among other value structures, he considers one value structure in which there is a special focus on the threshold problem. Note that here, the term threshold problem is assumed to contain the coordination problem. He finds that in package auctions, bidders usually submit a limited number of bids, leading to coordination becoming a critical issue: efficiencies are lower. This indicates that the coordination problem is correlated with the cognitive complexity issue.

Kagel et al. (2010) experimentally examine the performance of iterative ascending combinatorial clock auctions. They find that bidders usually bid on only a small number of packages, and tend to bid more often on their most profitable package than on their less profitable packages. Additionally, they find evidence that in 'hard' auctions (i.e. auctions with more items and more profitable packages) efficiency is lower than in 'easy' auctions (i.e. auctions with fewer items and less profitable packages). They use price feedback, and find evidence of the threshold problem: "*losing bidders could have possible obtained higher positive profits by continuing to bid*". Kagel et al. (2010) also check whether human bidder behavior can be simulated by automated bidders that bid myopically for the currently most profitable package, and find that such a straightforward bid simulator fares quite well in predicting auction outcomes. They extend their research for different value profiles in Kagel et al. (2014).

Scheffel et al. (2011) compare a nonlinear discriminatory price auction dubbed iBundle (for more information see Parkes and Ungar (2000)) to the Vickrey-Clarke-Groves auction and combinatorial auction formats with linear prices (ALPS and the combinatorial clock auction) in laboratory experiments. Two of the five value models that were used in their experimental design focus on what they call a threshold problem (which seems to reflect both the coordination and threshold problem). Scheffel et al. (2011) find no significant difference in efficiency across the different value models. However, the limited number of packages with positive private valuations could be an important factor in explaining that result.

Chernomaz and Levin (2012) consider single-round sealed-bid first price auctions with and without package bidding. They find that synergies decrease the profits of smaller bidders and increase profits of the large bidders. Furthermore, large bidders tend to bid more aggressively than smaller bidders. When synergies are low and package bidding is allowed, however, efficiencies go down due to the threshold problem.

Scheffel et al. (2012) study several combinatorial auction mechanisms. Specifically, they consider the HPB auction, the combinatorial clock auction, and the pseudo-dual price auction. One of their observations is that bidders select and evaluate most packages in the first round (package pre-selection). This accounts for most of the efficiency losses in their local synergies value model. Bidders turn out to select packages according to the relative valuations (i.e. the higher the valuations of the items are, the higher the likelihood that the bidders evaluate the package) and to evaluate only approximately 14 packages including two or more items (i.e. so-called *satisficing behavior*). If the hierarchy in HPB auctions does not fit the smaller bidders' preferences well, this often leads to smaller bidders failing to outbid a large bidder, which results in lower payoffs and efficiencies. Scheffel et al. (2012) conclude that the limited number of packages that bidders evaluate is a great barrier to efficiency, much more so than auction formats. They state that bidder decision support might play a big role in the design of large practical CAs research and design.

3.3 Notation and Terminology

In this section we provide basic terminology concerning combinatorial auctions in Section 3.3.1 (see also de Vries and Vohra (2003)). In Section 3.3.2, we describe so-called deadness and winning levels.

3.3.1 Basics

Consider a set $I = \{1, 2, \ldots, m\}$ of indivisible items which are auctioned using a first price (i.e. winning bidders pay the prices they bid) iterative combinatorial auction, and a set $A = \{1, 2, \ldots, n\}$ of bidders that participate in the combinatorial auction. Without loss of generality we assume that there are no multiples for items, or, equivalently, we assume that there is exactly one of each item. A bid b consists of three components: the bidder $a(b) \in A$ expressing bid b, the itemset $S(b) \subseteq I$ to which bid b applies, and the price that bidder $a(b)$ is expressing to pay for itemset $S(b)$ in bid b: $p(b)$. Hence, we see a bid b as a triple $(a(b), S(b), p(b))$, and we denote the set of bids by $\mathcal{B} = \{(a(b), S(b), p(b))|$ bidder $a(b)$ has expressed the willingness to pay $p(b)$ for itemset $S(b)\}$. Further, every bidder $a \in A$ has a nonnegative value $v_a(S)$ for every subset $S \subseteq I$; this value $v_a(S)$ is referred to as the *private valuation* of bidder a for the itemset S.

The well-known winner determination problem (WDP) is now the following: given the set of bids $\mathcal{B}$, determine the allocation of items to bidders that maximizes the sum of the values of the accepted bids, ensuring that each item is sold at most once. There is a straightforward integer programming formulation of this problem, that uses binary variables $x(b)$ which are equal to one if and only if bid $b \in \mathcal{B}$ is selected as a *winning* bid, meaning that bidder $a(b)$ receives itemset $S(b)$ at price $p(b)$.

$$\begin{array}{llll}
\text{(WDP)} & \max & \sum_{b \in \mathcal{B}} p(b)x(b) & \\
 & \text{s.t.} & \sum_{b \in \mathcal{B}:\, i \in S(b)} x(b) \leq 1 & \forall\, i \in I \\
 & & x(b) \in \{0,1\} & \forall\, b \in \mathcal{B}
\end{array}$$

Note that in this formulation of the WDP, a bidder can win multiple bids. This is the version of the WDP that is suited for an OR-bids bidding language (see e.g. Nisan (2000)). A well-known alternative to OR-bidding is XOR-bidding. An XOR-bids bidding language imposes a limit of at most one winning bid per bidder. The WDP formulation we present is easily adapted to allow for XOR-bids; one need only add an extra constraint for every bidder $a \in B$, namely that $\sum_{b \in \mathcal{B}: a(b)=a} x(b) \leq 1$.

Given all bids $b \in \mathcal{B}$, we can solve the resulting instance of WDP. This gives us optimal values for the decision variables, which we denote by $x(b)^*$ $(b \in \mathcal{B})$; the resulting optimum value is denoted by $WDP(I) = \sum_{b \in \mathcal{B}} p(b)x(b)^*$. The values of the decision variables correspond to a selection of winning bids that we denote by $\mathcal{B}^* = \{b \in \mathcal{B} | \ x(b)^* = 1\}$; in addition, we can identify a set $X = \{(a(b), S(b)) | \ b \in \mathcal{B}^*\}$, and we will refer to such a set X as an *allocation.*

The *value* of an allocation X depends on the private valuations of the bidders, and is denoted by $V(X) = \sum_{b \in \mathcal{B}^*} v_{a(b)}(S(b))$. This value can be seen as being distributed over the auctioneer on the one hand, and the bidders on the other hand. We use the term *auctioneer surplus* of an allocation X, denoted by $AS(X)$, to represent the revenue for the auctioneer: $AS(X) = \sum_{b \in \mathcal{B}^*} p(b) = WDP(I)$ (which indeed corresponds to the amount received by the auctioneer); we use the term *bidder's surplus* of an allocation X, denoted by $BS(X)$, as follows: $BS(X) = \sum_{b \in \mathcal{B}^*}(v_{a(b)}(S(b)) - p(b))$. It is easy to see that the total value of allocation is $V(X) = AS(X) + BS(X)$.

A particular allocation, called X^E, is found when each bidder bids his/her private valuation on each possible itemset, i.e., when each bidder $a \in A$ expresses a bid b on each itemset $S \subseteq I$ with price $p(b) = v_a(S)$. The value of this allocation, $V(X^E)$ is maximum over all allocations, and we use this quantity to be able to define the *economic efficiency* of any allocation X: $E(X) = \frac{V(X)}{V(X^E)}$. Notice that $0 \leq E(X) \leq 1$.

In words, economic efficiency measures how 'efficient' an auction is; it measures the total amount of achieved surplus relative to the maximum obtainable surplus. It represents a measure of social welfare. When efficiency is 100%, no participant in the auction, whether they are a bidder or the auctioneer, can improve their situation without making some other participant worse off. However, when efficiency is below 100%, there is still 'money left on the table'.

3.3.2 About deadness and winning levels

Observe that a bid $b \in \mathcal{B}$ in a combinatorial auction can be in one of three states (Adomavicius and Gupta, 2005), as depicted in Figure 3.1.

- Winning state: bid b is currently winning.
- Live state: bid b is currently not winning. However, it could become winning in a following round, depending upon the presence of new bids.
- Dead state: the bid is currently not winning, and has no chance of ever becoming a winning bid.

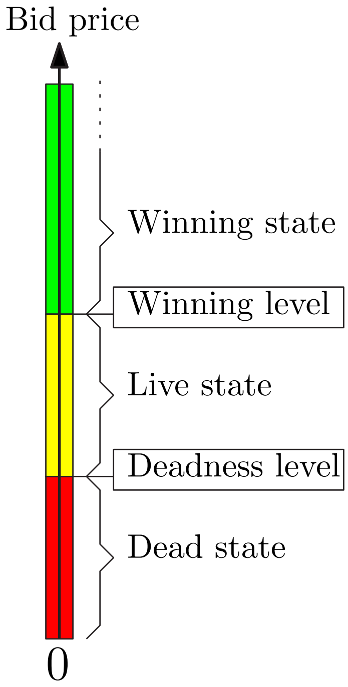


Figure 3.1: Bid states.

The state of a bid b depends on the corresponding price $p(b)$. Indeed, when we (imaginary) vary $p(b)$ from a low value, say 0, to a high value, the state of the bid will start in a dead state, next, at a specific value for $p(b)$ called the *deadness level*, the state will become live, and finally, at another specific value called the *winning level*, the state will become winning.

Definition 3.1. *The deadness level for a subset of items $S \subseteq I$, called $DL(S)$, is the minimum price, ceteris paribus, that some bidder $a \in A$ has to bid such that the resulting bid can become a winning bid in some future round.*

Deadness levels are calculated as follows. Consider a subset of items $S \subseteq I$. The objective value of the WDP induced by S (WDP-DL) corresponds to the deadness level of the itemset S:

$$\begin{array}{llll}
\text{(WDP-DL)} & \max & \displaystyle\sum_{b \in \mathcal{B}\colon S(b) \subseteq S} p(b)x(b) & \\
& \text{s.t.} & \displaystyle\sum_{b \in \mathcal{B}\colon i \in S(b)} x(b) \leq 1 & \forall\, i \in S \\
& & x(b) \in \{0,1\} & \forall\, b \in \mathcal{B} : S(b) \subseteq S
\end{array}$$

Thus, $DL(S) = \sum_{b \in \mathcal{B}\colon S(b) \subseteq S} p(b)x(b)^* = WDP(S)$. Deadness levels can be calculated after every round, for all subsets of items. Notice that deadness levels will only be different from zero when a set of bids is given, e.g. after a first round.

Definition 3.2. *The winning level for a subset of items $S \subseteq I$, called $WL(S)$, is the minimum price, ceteris paribus, that some bidder $a \in A$ has to bid on itemset S in order for that bid to become a winning bid in the next round.*

Winning levels are calculated as follows. Consider a set of items $S \subseteq I$, and consider the WDP restricted to itemset $I \setminus S$.

$$\begin{array}{llll}
\text{(WDP-WL)} & \max & \displaystyle\sum_{b \in \mathcal{B}\colon S(b) \subseteq (I \setminus S)} p(b)x(b) & \\
& \text{s.t.} & \displaystyle\sum_{\substack{b \in \mathcal{B}\colon S(b) \subseteq (I \setminus S) \\ \text{and } i \in S(b)}} x(b) \leq 1 & \forall\, i \in (I \setminus S) \\
& & x(b) \in \{0,1\} & \forall\, b \in \mathcal{B} : S(b) \subseteq (I \setminus S)
\end{array}$$

Thus, $WL(S) = WDP(I) - WDP(I \setminus S)$ for any itemset $S \subseteq I$. We denote the resulting set of winning bids by $\mathcal{B}^*(I \setminus S) = \{b \in \mathcal{B}|\ S(b) \subseteq (I \setminus S), x(b)^* = 1\}$.

Finally, we point out that the deadness and winning levels calculations as described in this section are suitable for OR-bids. In case of XOR-bids, these deadness and winning levels must be personalized. We refer to Petrakis et al. (2013) for information on how to adapt these calculations to also allow for XOR-bids.

3.4 Measuring Coordination and Threshold Problems

3.4.1 How to measure the coordination problem

In this section, we propose an index measuring the magnitude of the coordination problem in a combinatorial auction with private values. Recall from Section 1 that the coordination problem concerns the difficulty of identifying coalitions of bids that have the potential to become winning. Thus, we want to capture the difficulty that an individual bidder faces in order to identify those bids whose increase might lead to a larger bidder surplus. Let us first imagine that each bidder $a \in A$ identifies his/her most valuable itemset, i.e. the bidder identifies $S_a^* = \arg\max_{S \subseteq I} v_a(S)$. This is similar to a "straightforward bidding" strategy, which is a myopic bidding strategy in which bidders bid on the package that has the highest profit potential (Ausubel and Milgrom, 2002). Further, let us imagine that each bidder $a \in A$ bids a single bid; his/her private valuation, $v_a(S_a^*)$, on the itemset S_a^*, and nothing else. We call the allocation that is found after solving the corresponding WDP: X_1. In fact, generalizing, let us denote by X_k the allocation that results after solving the WDP when each bidder bids his/her true valuation on his/her k most valuable itemsets.

By construction, $V(X_k)$ is nondecreasing in k, and reaches $V(X^E)$ for a large enough value of k. Thus, an interesting value for k is the minimum value for which $V(X_k) = V(X^E)$. We denote this value by $k_{\max} = \min\{k : V(X_k) = V(X^E)\}$. This is an indication of the number of itemsets that each bidder should be prepared to bid on, starting from his/her most valuable itemset, to arrive at an efficient allocation. However, we remark that this does not imply that each bidder needs to express bids on his/her $k_{\max}$ most valuable itemsets in order to reach $V(X^E)$. In extremum it is e.g. possible that there is only one bidder that causes $k_{\max}$ to be greater than 1. In fact, one could calculate for every bidder $a \in A$ an individual k_a by looking at an efficient outcome and checking for every bidder how many packages, sorted from highest to lowest value, that specific bidder would have to consider to be able to arrive at an efficient allocation. If a bidder has multiple itemsets present in an efficient outcome, his/her k_a is determined by the least valuable of those itemsets. Then, as an alternative to $k_{\max}$, it can be interesting to examine the value of $\sum_{a \in A} k_a$.

We define the following index CI to measure the coordination problem:

Definition 3.3.

$$CI = \frac{\sum_{k=1}^{k_{max}} (V(X^E) - V(X_k))}{V(X^E)} = \sum_{k=1}^{k_{max}} (1 - E(X_k)).$$

A graphical interpretation of CI is depicted in Figure 3.2. Here, $E(X_k)$ is depicted for every value of k. For each $k < k_{\max}$, by definition, $E(X_k) < 100\%$. The surface of the red shaded area corresponds to the value of CI.

We give two examples to clarify index CI; one where there is essentially no coordination challenge and one where there is a large coordination challenge.

Example 1. *Consider the situation in Table 3.1, where there are three bidders (A, B and C) and two items (1 and 2). Clearly, $k_{max} = 1$ in this case, and hence $CI = 0$.*

Example 2. *Consider the private valuations in Table 3.2, where there are three bidders (A, B and C) and two items (1 and 2), and where ϵ represents an arbitrarily small amount greater than 0.*

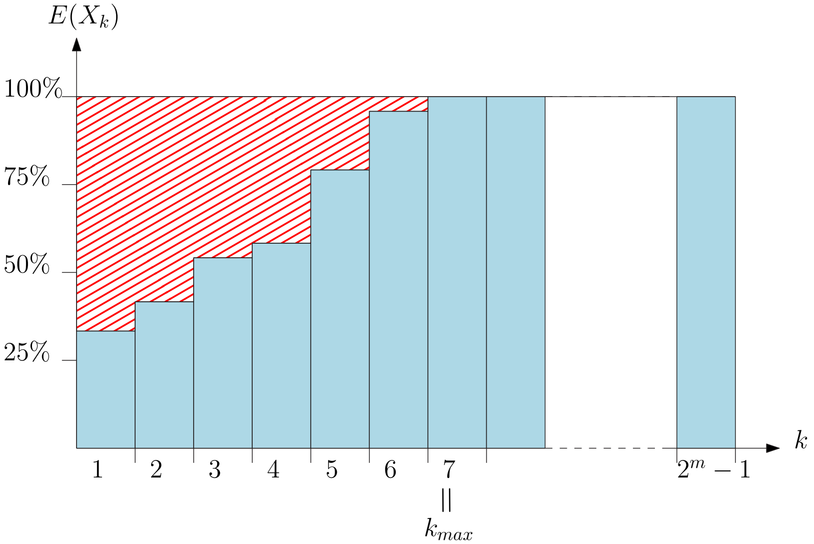


Figure 3.2: $E(X_k)$ graph.

	Package		
Bidder	{1}	{2}	{1,2}
A	10	10	40
B	10	10	25
C	10	10	25

Table 3.1: Private valuations that lead to no coordination challenge.

	Package		
Bidder	{1}	{2}	{1,2}
A	15	25	50+ϵ
B	35	50	50+ϵ
C	50-ϵ	50	50+ϵ

Table 3.2: Private valuations that lead to a difficult coordination challenge.

In this case, $k_{max} = 3$*. In other words; there is a bidder that has to bid on his/her three highest value packages in an auction where there are a total of three possible packages. However, the private valuations are such that a significant part of the possible surplus will be missed in* $V(X_1)$ *and* $V(X_2)$*. Specifically,* $V(X_1) = 50 + \epsilon$*,* $V(X_2) = 50 + \epsilon$*, and* $V(X^E) = V(X_3) = 100 - \epsilon$*. This leads to CI* ≈ 1 *when* ϵ *approaches 0.*

Let us discuss some properties of CI.

- CI ≥ 0, and the larger CI, the larger the coordination challenge present in that auction.
- CI is an index that corresponds to a set of valuations as a whole. It does not depend on actual bids, nor on a particular allocation. CI does require knowledge of the private valuations, which on the one hand may be considered demanding in non-experimental settings, but on the other hand seems necessary knowledge in order to be able to quantify the coordination problem at all.
- Items that do not have a positive marginal value to any bidder, do not affect CI.
- CI takes the number of possible packages, the number of bidders, and the diversity of their private valuations, into account. Intuitively, one would expect the diversity of the preferences to be present in any measure for the coordination problem. The values $V(X_k)$ achieve exactly this.

Concluding, a potential attractive use of index CI lies in the design of a particular combinatorial auction. A relatively high value for CI indicates that bidders may not have diverse preferences, and that attention should be paid to relaxing the cognitive burden bidders face, e.g. using bidder support in the shape of feedback. Alternatively, if the index is estimated to be relatively low, no special attention is necessary to guide the bidders. Finally, in many experiments on combinatorial auctions, the participants are equipped with a particular private valuation. As a result, one can compute CI, and explain the results in the light of the value found for CI. This allows us to compare different auctions. As an example, we did the calculations for the threshold value model found in Bichler et al. (2017) on ten randomly generated instances, and found an average CI of 0.42, an average $k_{\max}$ of 3.70, and an average $\sum_{a \in A} k_a$ of 5.30.

3.4.2 How to measure the threshold problem

In this section, we propose an index measuring the severity of the threshold problem. The threshold problem concerns the difficulty of overcoming the gap for a particular itemset S between the current (nonwinning) bid b on itemset S, and its winning level. Notice that whereas the index CI applies to an auction, and is independent of the particular bids, the index we propose for the threshold problem depends on the bids, and applies to every itemset S, and any set of bids $\mathcal{B}$.

More concrete, consider a set of non-overlapping, non-winning bids $\mathcal{B}_L$, and let $S = \bigcup\limits_{b \in \mathcal{B}_L} S(b)$. When computing the winning level of itemset S by solving the corresponding IP, we find $WL(S)$, and, in addition, we find a new set of winning bids called $\mathcal{B}^*(I \setminus S)$. We are interested in bids that are present in $\mathcal{B}^*(I \setminus S)$ and not present in $\mathcal{B}^*$. Indeed, these bids can be seen as the 'newcomers' in the best-possible allocation that includes $\mathcal{B}_L$. We denote this set of newcomer bids by $N(\mathcal{B}_L) = (\mathcal{B}^*(I \setminus S) \cup \mathcal{B}_L) \setminus \mathcal{B}^*$. We propose the following index to capture the threshold problem a set of non-overlapping, non-winning bids $\mathcal{B}_L$ faces.

Definition 3.4. *The* threshold index *TI of a set of non-overlapping, non-winning bids $\mathcal{B}_L$, and with $S = \bigcup\limits_{b \in \mathcal{B}_L} S(b)$, equals*

$$TI(\mathcal{B}_L) = \frac{WL(S) - \sum_{b \in N(\mathcal{B}_L)} p(b)}{\sum_{b \in N(\mathcal{B}_L)} (v_{a(b)}(S(b)) - p(b))}.$$

An informal explanation of this index is that the burden of reaching the winning level needed to turn bid $b \in N(\mathcal{B}_L)$ into a winning bid can be distributed over all bids that would profit from this. These bids that profit are exactly the newcomers, i.e. $N(\mathcal{B}_L)$, and it is not difficult to see that the threshold index $\mathrm{TI}(\mathcal{B}_L \cup N(\mathcal{B}_L))$ is in fact equal to $\mathrm{TI}(\mathcal{B}_L)$. Clearly, if $\mathrm{TI}(\mathcal{B}_L)$ equals zero there is no threshold to overcome. In that case, the current coalition is equivalent to the winning one revenue-wise, but might not be winning because of some auction rule. If $\mathrm{TI}(\mathcal{B}_L)$ is between zero and one, there is a threshold to overcome. However, given the existing private valuations that correspond to the non-winning bids, this threshold can be overcome. The closer $\mathrm{TI}(\mathcal{B}_L)$ is to one, the higher the losing coalition members need to bid in relation to their respective private valuations and the more difficult it is to overcome the threshold. If $\mathrm{TI}(\mathcal{B}_L)$ equals one, the current losing coalition can at most 'match' the current winning one, and this can only happen when they bid their private valuations. Finally, if $\mathrm{TI}(\mathcal{B}_L)$ is greater than one, the threshold is too large to overcome. Thus, excepting cases where bidders place bids that exceed their private valuations, this coalition can never outbid the current one. Remark that the denominator can equal zero if and only if all bidders in the losing coalition entered bids equal to their respective private valuations. If this is the case, then there is also an insurmountable threshold problem. We give a comprehensive example of the TI below.

Example 3. *Consider the private valuations in Table 3.1. Suppose that in some round the following bids are made:* $(A, \{1, 2\}, 15)$, $(B, \{1\}, 5)$, *and* $(C, \{2\}, 5)$. *Clearly, the bid by bidder A becomes winning and the bids by bidder B and C are losing. The TI value for the coalition of non-winning and non-overlapping bids by bidders B and C is then* $\frac{15-5-5}{5+5} = 0.5$. *Since* $TI < 1$, *the threshold problem can still be overcome.*

Next, suppose that in some round the following bids are made: $(A, \{1, 2\}, 20)$, $(B, \{1\}, 5)$, *and* $(C, \{2\}, 10)$. *Clearly, the bid by bidder A becomes winning and the bids by bidder B and C are losing. The TI value for the coalition of non-winning and non-overlapping bids* *by bidders B and C is then* $\frac{20-5-10}{5+0} = 1.00$. *Since TI* $= 1$, *bidders B and C can at most 'match' the current winning bid of 20 by bidder A if they bid equal to their respective private valuations.*

Finally, suppose that in some round the following bids are made: $(A, \{1,2\}, 22.5)$, $(B, \{1\}, 7.5)$, *and* $(C, \{2\}, 7.5)$. *Clearly, the bid by bidder A becomes winning and the bids by bidder B and C are losing. The TI value for the coalition of non-winning and non-overlapping bids by bidders B and C is then* $\frac{22.5-7.5-7.5}{2.5+2.5} = 1.5$. *Since TI* > 1, *the threshold problem becomes insurmountable; bidders B and C, providing they bid rationally in the sense that they do not bid higher than their respective private valuations, can never outbid bidder A.*

We remark that this index can be seen as a generalization of the 'gain' concept, as used in Banks et al. (2003). In Banks et al. (2003) the gain effect measures "*the relative value of the optimal allocation (V^*), which is purposefully composed of several small bidder packages, and the next highest value allocation (V), which is constructed to be a single bidder's value for the large package covering an optimal set of packages.*" The relation with our index TI is as follows. Given a set of private valuations, first identify the two highest value allocations, say X_1 is the highest value allocation and X_2 is the second highest value allocation. Next, if the bidders in X_1 bid truthfully, i.e. a bid price equal to their private valuations, and the bidders in X_2 bid nothing, we can calculate TI(X_2). It is not difficult to see that the gain is now equal to the value of TI(X_2).

Finally, we point out some other uses of TI. One significant application lies in the possibility of creating feedback for a losing coalition of bids. The numerator of TI can be calculated without information on private values, and corresponds to the threshold faced at a specific time by those losing bids. This number can be used e.g. for concrete bid price suggestions of price rules. We apply TI to that end in Section 3.5.2. Another application of TI lies in the generation of private value sets for e.g. laboratory or simulation experiments. Using TI in the gain-manner, it is possible to design private values that induce a certain level of threshold problem. We use TI in that way in Section 3.6.2. Finally, the 'gain'-manner of using TI can also be used to compare expected threshold problems. As an example, we did the calculations for the threshold value model found in Bichler et al. (2017) on ten randomly generated instances, and found gains varying from 1.013 to 1.203, with an average gain of 1.099.

3.5 Coalitional Feedback

3.5.1 Factual Coalitional Feedback

Factual coalitional feedback (FCFB) is a new type of feedback that is designed to help overcome coordination problems in CAs. As stated earlier, in CAs bidders first face coordination problems that can negatively impact efficiencies and revenues. The novelty of our proposed coalitional feedback lies in the information we provide regarding how many other bids can help each other to become (provisionally) winning. This will give bidders an idea whether or not the coordination problem, keeping in mind their own valuations, can be overcome. It solves the question *"are there other bids that complement my bid?"*

We now describe how to obtain factual coalitional feedback. First, consider a non-winning bid $b \in \mathcal{B}$. When calculating the winning level for b, i.e. $\mathrm{WL}(S(b))$, we also find a coalition of bids that are 'newcomers', i.e. bids that were not winning before but become winning together with b. This coalition is denoted by $N(b) = b \cup \mathcal{B}^*(I \setminus S(b)) \setminus \mathcal{B}^*$, and the number of bids in that coalition is $|N(b)|$. Next, if $|N(b)| > 1$, the following message goes out to all bids in $N(b)$: "If $|N(b)|$ bids, including this one, are collectively raised by $(\mathrm{WL}(S(b)) - p(b))$, these $|N(b)|$ bids become winning." It is important to see that all bidders in $|N(b)|$ face the same 'increment', i.e. $(\mathrm{WL}(S(b)) - p(b))$. Factual coalitional feedback is most relevant when there are multiple bids that can become winning together, thus beating the currently winning bids, given that all bids outside of the coalition remain the same. Note that it is possible to receive multiple such messages for a single bid. A single bid receives such a feedback message each time it appears in an allocation that makes some non-winning bid winning. Clearly, it is a potential remedy against coordination problems, as bidders can now consider the number of messages they receive, the suggested 'increments' for each of those suggestions coupled with the size of the coalition, and their own private valuation.

We illustrate this using an example.

Example 4. *Consider an auction with 4 bidders and 3 items, and the set of bids presented in Table 3.3. A * in the deadness level column indicates the bid currently is not dead (i.e. the bid is live). A * in the winning level column indicates the bid is currently winning.*

b	$a(b)$	$p(b)$	$S(b)$	DL($S(b)$)	WL($S(b)$)
1	1	2	{2}	8	16
2	1	5	{1}	5*	13
3	1	25	{1,2,3}	25*	25*
4	2	4	{1}	5	13
5	2	4	{2}	8	16
6	2	10	{1,2}	13	21
7	3	8	{2}	8*	16
8	4	4	{3 }	4*	12

Table 3.3: A set of bids and their corresponding deadness levels and winning levels.

Consider the bids in Table 3.3. Bid number 2 will receive the following factual coalitional feedback: "If 3 bids, including this one, are collectively raised by 8, these 3 bids become winning." It is not hard to see that the coalition induced by bid number 2 consists of bid numbers 2, 7 and 8. In fact, bid numbers 7 and 8 will receive the same message that bid number 2 will. Also note that the winning level for bid 8 is 12, which might be too high for bidder 4. The suggested coalition indicated that the increment, 8, could be split up over a total of 3 bids.

3.5.2 Suggestive Coalitional Feedback

Suggestive coalitional feedback (SCFB), goes one step further than FCFB, and adds a concrete bid suggestion in addition to the feedback given with FCFB. As such, SCFB is designed to combat both coordination and threshold problems. With SCFB, similar to FCFB, a bid b entered by bidder $a(b)$ on a set of items $S(b)$ will receive feedback in the following manner: "If $|N(b)|$ bids, including this one, are collectively raised by $(\mathrm{WL}(S(b)) - p(b))$, these $|N(b)|$ bids become winning. We suggest you bid $p(b) + \frac{(\mathrm{WL}(S(b))-p(b))}{|N(b)|}$." The same message also goes out to the other bidders in $N(b)$. Suggestive coalitional feedback can appear if there are multiple bids that can become winning together, thus beating the currently winning bids, given that all bids outside of the coalition remain the same. It solves the questions *"are there other bids that complement my bid?"* and *"what price should I bid, so that I become winning instead of the currently winning bid(s)?"* There are many ways to give a concrete bid suggestion and they all have advantages and disadvantages. For example, one could look at the bid amounts and suggest an amount proportional to that. The idea is then to suggest a higher bid price to bids that are already higher. Large bidders, i.e. bidders with relatively high private values, then get higher suggestions. The disadvantage of this approach is that if smaller bidders, i.e. bidders with a relatively low private values, that bid relatively high compared to their private values, can get a suggestion that is too high. Another approach is to take into account the number of items in the bids, or even incorporate the Shapley value (Bichler et al., 2017). However, these sort of technicalities often make it unnecessarily difficult for bidders to understand what is going on in the feedback. For that reason, we opted for a concrete suggestion that is *fair* in the sense that the increment is divided equally among the bidders in the coalition. Note that it is again possible to receive multiple such messages for a single bid. A single bid receives such a feedback message each time it appears in an allocation that makes some non-winning bid winning. Also note that it is possible that the concrete bid suggestion exceeds a bidder's private valuation on a subset of items, i.e. $p(b) + \frac{(\mathrm{WL}(S(b)) - p(b))}{|N(b)|)} > v_a(S)$. However, in practice this is unavoidable; the auctioneer does not know the private valuations.

Finally note that coalitional feedback, both the factual and the suggestive variants, give no concrete suggestion as to which package of items to bid on. Instead, it takes into account previously made bids, and informs the bidder about coordination opportunities, and in case of suggestive coalitional feedback adds a price suggestion. In other words: bidders still need to find packages that are of interest to them (where they can get a positive bidder surplus), but coordination and cooperations with other bidders becomes easier.

Example 5. *Consider again the bids in Table 3.3. Bid number 2 will receive the following factual coalitional feedback: "If 3 bids, including this one, are collectively raised by 8, these 3 bids become winning. We suggest you bid 8." It is not hard to see that the coalition induced by bid number 2 consists of bid numbers 2, 5 and 8.*

3.6 Methodology

3.6.1 Experimental environment

To experimentally study the effect of feedback on the ability for bidders to overcome coordination and threshold problems, we set up iterative CAs in a lab[1] using the z-Tree software (Fischbacher, 2007). In these auctions, bidders compete to acquire a number of items, and are allowed to bid on any subset of the items. We impose no limit on the number of bids that a bidder can submit, nor do we impose any activity rules. We use a minimum bid increment of 1, and only allow bids lower than or equal to the relevant private valuation. This eliminates gaming behavior; bidders can no longer

[1] The laboratory experiment has been approved by the Social and Societal Ethics Committee (SMEC) of the KU Leuven.

incur possible losses. More details on how the private valuations were set are given in Section 3.6.2.

We opted for an OR-bids bidding language, which means that given a number of bids from a bidder, the auctioneer can accept any non-overlapping set of these bids and charge the sum of the specified prices (see e.g. Nisan (2000)). The XOR-bidding language would be a more expressive alternative, but, as stated in both Brunner et al. (2010) and Scheffel et al. (2012), XOR-bidding can lead to problems if bidders only submit few bids. Indeed, a limited number of bids quickly leads to a number of unsold items, which may have a considerable impact on efficiency. Since we use superadditive valuations in our laboratory experiments, the OR-bids bidding language is well suited.

The auction proceeds in rounds, until two consecutive rounds occur in which the total auction revenue does not increase compared to the previous round. In other words, if three consecutive auction rounds lead to the same revenue, the auction closes. When that happens, the provisionally winning allocation becomes the final winning allocation. This closing rule effectively eliminates sniping strategies, where bidders suddenly make (higher) bids in the last round. Note that there is a potential tradeoff present between our closing rule and the auction duration. However, in our experiments we encountered no such adverse effects.

3.6.2 Experiment factors

In this section we discuss the different factors (independent variables) in our experimental design. The experimental design is given in the Appendix in Table C.1.

Item and bidder structure factor

We use four different item/bidder structures, as shown in Figures 3.3a and 3.3b. The item structures are similar to settings in Kazumori (2010), Scheffel et al. (2011), and Vangerven et al. (2017b). Note that the structure corresponds to the geometrical setting discussed in Chapter 2, specifically to a 2- or 3-row problem. Bidders need not bid on sets of adjacent items, however, their valuations (see Section 3.6.2) are such that if complementarity effects exists, they involve adjacent items.

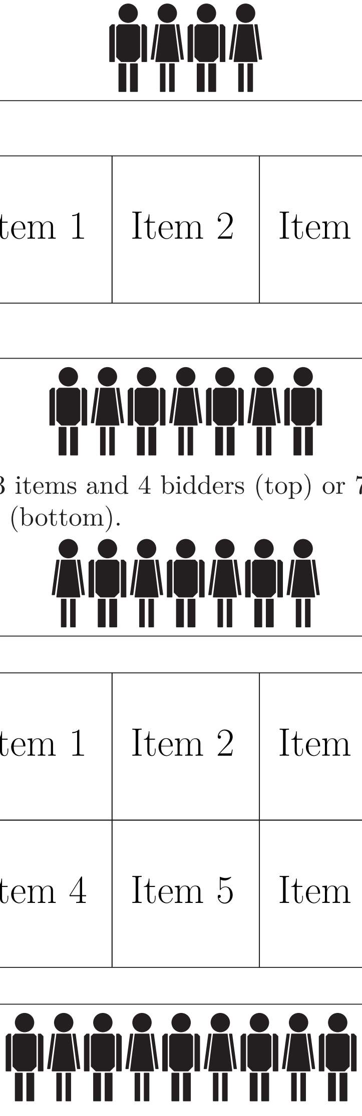
Combining both the number of items and bidders, we obtain what we refer to as the factor *structure*. The factor structure has four levels: 3 items with 4 bidders (STR1), 3 items with 7 bidders (STR2), 6 items with 7 bidders (STR3), and 6 items with 9 bidders (STR4).

We remark that subjects are randomly assigned to an auction, and stay in the same level of the factor structure during four consecutive auctions. In other words: four consecutive auctions in a session with the same level of structure have the same subjects.

Feedback factor

The second factor in our laboratory experiments is feedback. Feedback is calculated by the auctioneer after each round and communicated to the bidders. We use a hierarchy of feedback involving four levels, as depicted in Figure 3.4. The first (i.e. lowest) level, *outcome feedback* (FB1), consists of showing the (provisionally) winning allocation along with the prices corresponding to that allocation. Without this information, it would make little sense to hold an iterative combinatorial auction. The second feedback level (FB2) consists of the feedback given in FB1, but adds winning and deadness levels for any set of items. We call this *bid states* feedback. Another layer up the hierarchy (FB3), we add *factual coalitional feedback* on top of the feedback given in FB2. The fourth feedback level (FB4) adds a concrete bid suggestion, *suggestive coalitional feedback*, in addition to the feedback given in FB2. Any feedback regarding a particular subset of the items is only displayed if the bidder clicks on that subset. Note that we are able to calculate all feedback in a fraction of a second, even for the auctions with 6 items and 9 bidders.

Our coalitional feedback bears some resemblance to the price



(a) 3 items and 4 bidders (top) or 7 bidders (bottom).

(b) 6 items and 7 bidders (top) or 9 bidders (bottom).

Figure 3.3: Item and bidder structure.

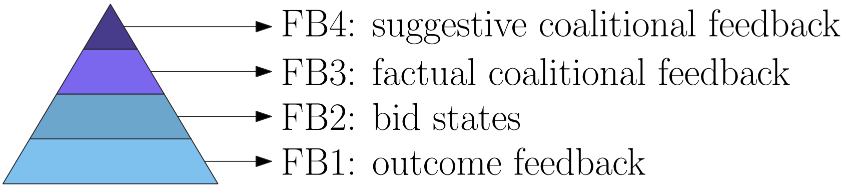

FB4: suggestive coalitional feedback
FB3: factual coalitional feedback
FB2: bid states
FB1: outcome feedback

Figure 3.4: A combinatorial auction feedback hierarchy.

rule based on coalitional winning levels used in Bichler et al. (2017). There are however a number of important differences on the implementation level. (i) In Bichler et al. (2017), the coalitional pricing rule is calculated for currently losing bidders, but in their experiments the price rule for a bid follows from one coalition, specifically the coalition with the lowest price suggestion for that bid. In our experiments, we allow for multiple messages to be displayed, because it is possible that the lowest suggested amount corresponds to a coalition that faces an insurmountable threshold. Moreover, we want to make sure that when one coalition member receives a price suggestion, all other members whose collaboration is required also receive this suggestion. (ii) Another difference is that in our versions of coalitional feedback we only consider coalitions of live bids for feedback. Disregarding coalitions that include dead bids has the advantage of limiting the number of feedback suggestions, and requiring bidders to first bid past their respective deadness levels. It also encourages activity in the auction, without actually requiring an explicit activity rule.

Coordination & threshold: CT factor

With the coordination & threshold (CT) factor, we aim to design private valuations leading to specific settings for the coordination and threshold problem[2]. However, it is not trivial to a priori create instances that have a certain level of threshold difficulty. Indeed,

[2] The private valuations used in the experiment are available here: `https://feb.kuleuven.be/public/u0093797/Valuations/`.

contrary to the CI, which only depends on bidder valuations, the TI is dependent on bidder behavior during the auction, which can be steered through the valuations each bidder receives only to some extent.

Each structure has a number of so-called small bidders who are interested in different items and 1 large bidder who is mainly interested in a package containing all items. Let us first discuss how the valuations for ‘small bidders’ are generated. Each small bidder has one favorite item: in all structures but STR2, this bidder’s valuation tops the valuations of all other bidders for this item (in STR2, every item has two small bidders who are specifically interested in that item). The valuations of the other individual items depend on how close they are to that item. In Figure 3.5 we give an example of the single item valuations of a small bidders in STR3, which has 6 items for sale. Clearly, that bidder is mainly interested in item 1 (in the top left corner), for which he/she has a valuation of 20. Valuations for adjacent items decrease by 50% with each step they are further away from the item of main interest. The valuations for the small bidders are purely additive. For example, the bidder in Figure 3.5 would have a valuation of 30 for items 1 and 2 combined, and a valuation of 13 for items 4 and 6 combined.

Item 1 20	Item 2 10	Item 3 5
Item 4 10	Item 5 5	Item 6 3

Figure 3.5: Example of valuations of a small bidder in STR3.

Now, the valuations for the large bidder can be generated. Depending on the ratio W of the sum of the highest individual valuations of the small bidders and the large bidder's valuation for the complete set of items, we discern three levels for the coordination/threshold factor.

- CT1: $W \in [92\%, 94\%]$. Since the highest valuation is that of the large bidder for the complete itemset, it is logical that $k_{\max}$ equals 1 and CI equals 0. In other words, there is no real coordination problem in these instances. On the other hand, we expect coalitions of small bidders to face an insurmountable threshold problem. Still, it remains interesting to see how far the small bidders will drive up the price for the large bidder.
- CT2: $W \in [105\%, 107\%]$. For these instances, CI values are around 0.20 for STR1-2 and 3.93 for STR3-4. The values for $k_{\max}$ vary around 4.5 for STR1-2 and around 42.38 for STR3-4. In other words, CT2 represents an easy coordination problem. Coalitions of smaller bidders have a small advantage over the large bidder; we expect a difficult threshold problem.
- CT3: $W \in [123\%, 125\%]$. These instances have a more pronounced coordination problem, with CI values around 0.63 for STR1-2 and 7.73 for STR3-4. Values for $k_{\max}$ vary around 4.5 for STR1-2 and around 49.5 for STR3-4. However, as the valuation of the optimal coalition amply exceeds the valuation of the large bidder, we anticipate an easy threshold problem.

Given the large bidder's valuation for the complete itemset, using super-additivities of 20% for every additional adjacent item, we can work backwards to find the valuations for all possible subsets of items.

Note that STR4 is somewhat more complex, since there are 9 bidders participating in this structure level. Similar to STR3, we have 6 small bidders, interested mainly in one single item, and 1 large bidder, interested in the package containing all items. However, there are also medium bidders, who are mainly interested in either the top row (items 1, 2, and 3) or bottom row (items 4, 5, and 6) and who have super-additivities only for adjacent items of specific interest on that row. This adds an extra layer of competition and complexity to the combinatorial auction.

We created two sets of private values for every combination of the factors Structure and CT, leading to a total of 24 different sets of private values.

Finally, we remark that participants are not told their roles (e.g. small bidder), but rather have to discover each other's valuations in the auctions. The values are private, but participants are informed that all values are at least additive. Participants take part in consecutive auctions, but valuations (roles) rotate between these auctions.

3.7 Results

In the following we report the results from laboratory experiments that were carried out at KU Leuven. A total of 192 auctions were held, and 324 subjects participated. Participants were students at the Faculty of Economics and Business. A printout of the instructions was handed out to every participant in the beginning of the experiment. All participants worked their way through the instructions and filled in a set of test questions. Participants were free to ask questions. Once all subjects were done filling in the test questions, and when all those questions were answered correctly, the auctions started. In every session, the same experimenter was present, and all the experiments were held in the same room. Students received a bonus point on the exam of a course they had to take for showing up, and a monetary incentive that depended on performance in the auctions. Performance is measured by the difference between the private values and the prices paid for the final winning bids. On average, the participants earned 9.62 euros.

3.7.1 Validation

Before we start discussing the results of the experiment, it makes sense to first validate whether the experimental design resulted in what we thought it would. Hence, we first examine the realized TI values, followed by an exploration of how often bidders followed the suggestive coalitional feedback (FB4).

The realized TI values

In order to validate whether the private valuations generated for the experiments indeed lead to the threshold problems we expect them to, we first investigate the realized TI values. Recall that the TI is not a single number for each auction. We calculate the TI for specific coalitions. In STR1 and STR2, we look at the highest value coalition of 3 small bidders versus the large package bidder. In STR3, we look at the highest value coalition of 6 small bidders versus the large package bidder. In STR4, we look at the highest value coalition of small/medium bidders versus the medium/large bidder(s). In other words, the TI values are calculated from the perspective of a coalition of smaller bidders. We take the average for every auction that has such a TI value (note that not all auctions have TI values to compute if the coalitions mentioned above were not present) for every level of CT, and present the results in Table 3.4.

CT	Mean
1	0.98
2	0.25
3	0.11

Table 3.4: Mean TI values per level of CT.

Overall, it does appear that threshold problems we wanted to create, were indeed present. For CT1, we see a TI value that, on average, is almost equal to 1. Considering that there were auctions in which the coalition of smaller bidders all bid up to their private valuation but did not win, and hence did not have a TI value, this TI value indeed seems to correspond to insurmountable threshold problems for a coalition of small bidders. For CT2, i.e. difficult threshold problems, we find an average TI value of 0.25, and for CT3, i.e. easy threshold problems, we find a value of 0.11. This also looks to be in line with expectation; more difficult threshold problems have a higher average TI value.

Do bidders follow FB4?

When introducing non-binding price suggestions, like we do with our suggestive coalitional feedback, it is evident that it is interesting to check whether bidders actually use those suggestions. To that end, we compute a number of interesting statistics and present them in Table 3.5. In that table, w corresponds to the average number of rounds a bidder received at least one suggestion, x corresponds to the average number of rounds where a bidder follows at least one suggestion by bidding at least the suggested amount (provided that the suggestion was below the relevant private valuation), and y corresponds to the average number of rounds a bidder followed a suggestion by bidding up to their private valuation (provided that the suggestion was above the relevant private valuation).

	CT1	CT2	CT3
w	4.15	3.95	4.20
x	0.85	1.61	1.16
y	0.93	1.62	1.30
$(x+y)/w$	0.43	0.82	0.59

Table 3.5: FB4 usage.

It becomes clear that on average there were quite a few rounds where bidders received a suggestion (w), whatever level of CT is applied. However, things become more interesting when looking at the x and y statistics. This leads to the following observation.

Observation 3.1. *When confronted with difficult threshold problems (CT2), bidders follow bid price suggestions to a larger extent than they do when confronted with easy (CT3) or insurmountable (CT1) threshold problems.*

A possible explanation for bidders being less inclined to follow suggestions, is that the suggestions can go out to bidders that already have a (provisionally) winning bid. In that case, bidders might not follow the suggestions in hopes of winning at a low price. Bidding on more packages might not be in the best interest of the bidder, because that could also drive up prices for other preferred packages. We remark here that, although it seems that bidders often bid in line with the suggestions, this does not automatically need to lead to higher efficiencies or revenues. Indeed, there is, e.g. still a possibility that the coordination problem is not solved, or the threshold problem is still not overcome (not all bidders follow the suggestions). Another explanation, in the case of insurmountable threshold problems, is that the suggestions coalitions of losing bids/bidders receive are simply too high. In our experiment, the upper bound on the bid price is the private value. With suggestions higher than private valuations, bidders could just stop bidding altogether.

3.7.2 Market outcomes: efficiency (E(X)) and auction revenue (AS(X))

Efficiency results

Recall from Section 3.3 that economic efficiency is a measure of social welfare: it measures how much of the total possible surplus is obtained by the combinatorial auction. Tables 3.6a—3.6b contain the mean efficiencies per level of STR, CT and FB. Overall, efficiencies were quite high.

In addition to examining the means, we looked at the box plots[3]

[3] Not all software packages draw box plots in the same manner, e.g. R has no less than 9 different quantile method variants. The quantiles in this dissertation are calculated as follows: given a sorted sample of a distribution, say

	STR1	STR2	STR3	STR4
FB1	0.993	0.978	0.963	0.978
FB2	0.990	0.999	0.986	0.992
FB3	0.991	0.986	0.974	0.992
FB4	0.983	0.995	0.990	0.972

(a) Mean E(X) for factors FB and STR.

	CT1	CT2	CT3
FB1	0.986	0.977	0.968
FB2	0.999	0.982	0.998
FB3	0.986	0.989	0.980
FB4	0.991	0.985	0.978

(b) Mean E(X) for factors FB and CT.

Table 3.6: Mean E(X).

of efficiency per level of feedback, see Figure 3.6. Here we see that in the cases where only basic feedback is given (FB1) efficiencies show the highest degree of dispersion. This indicates that simply showing the (provisionally) winning allocation as feedback is often insufficient for bidders to find an efficient outcome. The difference between FB1 on the one hand, and FB2, FB3, and FB4 on the other hand is striking: it seems deadness and winning levels are important in guiding bidders to an efficient outcome. The FB2, FB3 and FB4 box plots look quite similar, though FB3 and FB4 show a couple of outliers.

Our data is not normally distributed, so we use the non-parametric

$X_1 < \ldots < X_N$, for any real number p with $0 \leq p \leq 1$ the "p"-quantile is defined as $X_p = X_{N \cdot p}$ if $N \cdot p$ is an integer number, and $X_p = \frac{1}{2}(x_{\lfloor N \cdot p \rfloor} + x_{\lceil N \cdot p \rceil})$ if $N \cdot p$ is not an integer. For clarity, the box plots we use in this dissertation use the 0.25-quartile of the data as the lower quartile, the 0.5-quartile as the median, the 0.75 quartile as the upper quartile, and use as a lower (higher) whisker the smallest data value which is larger than the lower quartile - (+) 1.5 times the interquartile distance.

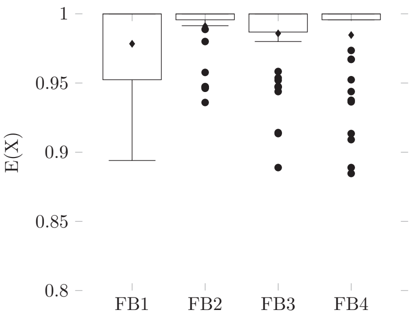


Figure 3.6: Box plots of FB.

Wilcoxon-Mann-Whitney test to examine the differences in efficiencies. The notation $\prec$, $\prec^*$ and $\prec^{**}$ respectively denote a difference at 10%, 5% and 1% significance level, and $\approx$ denotes we cannot reject the null hypothesis.

Observation 3.2. *Efficiencies ranked by Wilcoxon-Mann-Whitney tests:*

$$\mathit{FB1} \prec^{**} (\mathit{FB2} \approx \mathit{FB3} \approx \mathit{FB4})$$

At .01 significance level, we conclude that the efficiencies obtained under FB1 are lower than those obtained under FB2, FB3, and FB4. At .1 significance level, we cannot reject the hypothesis that the efficiencies obtained under FB2, FB3, and FB4 come from the same distribution.

Figure 3.7 depict the percentage of auctions that ended efficiently. FB1 shows the highest percentage of non-efficient auctions. The number of non-efficient auctions with FB2, FB3, and FB4 is lower than FB1, but is not zero.

We now examine whether the factor structure has an impact on efficiency. Since the separate box plots of STR1 and STR2 were remarkably similar, we combined the STR1 and STR2 data. The STR3 and STR4 box plots were almost identical as well, hence we combined that data too. The resulting box plots can be found in Figure 3.8. It appears that in STR1 and STR2, i.e. environments

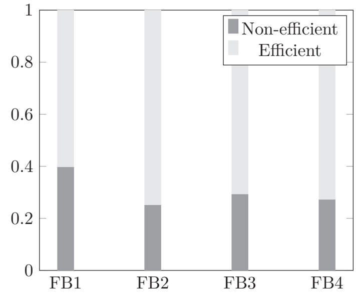


Figure 3.7: Percentage of efficient and non-efficient auctions.

with a limited number of items (three), efficiencies are quite high. While there are a number of outliers, the degree of dispersion is very limited. Looking at the more challenging environments, namely STR3 and STR4 where there are 6 items on auction, the spread is larger. However, efficiencies remain high.

Observation 3.3. *Auctions with fewer items lead to higher efficiencies.*

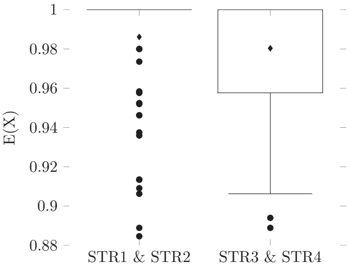


Figure 3.8: Box plots of FB per level of structure.

While the structure does seem relevant for auction outcomes, it is perhaps more interesting to see whether the level of the CT factor has an impact. To that end, we constructed box plots for efficiency per level of FB for the different levels of the CT factor in Figure 3.9. In the insurmountable threshold case, CT1, we see that efficiencies are almost always 100%, i.e. the large bidders win when they should win. In the difficult threshold case, CT2, we see that the degree of dispersion for FB1 and FB2 is quite large. However, FB3 and FB4 fare better than FB1 and FB2. We also see that there is more variation in efficiency in the CT2 case compared to both CT1 and CT3. This is an indication that the threshold problem indeed manifests itself here; the efficient allocation is not always reached. Even in the easy threshold case, CT3, we see that the efficient allocation is not always reached. FB1 shows a large spread, whereas FB2 shows no spread at all. FB3 and FB4 are situated in between FB1 and FB2.

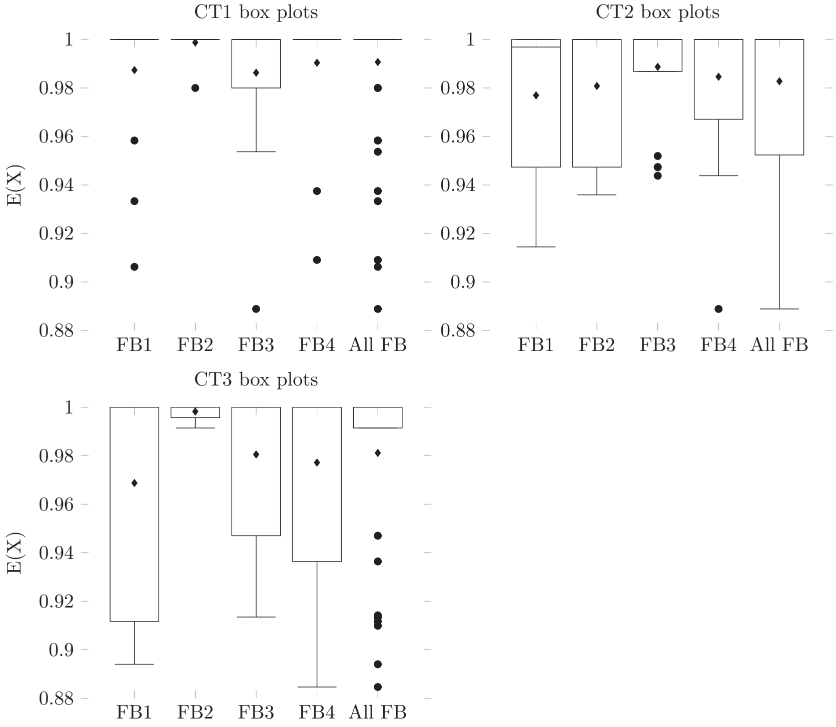


Figure 3.9: Box plots of FB per level of threshold.

Observation 3.4. *Table 3.7 contains the efficiencies ranked by Wilcoxon-Mann-Whitney tests.*

CT	Wilcoxon-Mann-Whitney tests
1	FB1 $\approx$ FB2 $\approx$ FB3 $\approx$ FB4
2	(FB1 $\approx$ FB2) $\prec$ (FB3 $\approx$ FB4)
3	FB1 $\prec$ FB2 FB1 $\approx$ FB3 $\approx$ FB4 FB4 $\prec$ FB2

Table 3.7: Ranked efficiencies per level of CT.

For CT1, we cannot reject the hypothesis that the efficiencies obtained under FB1, FB2, FB3, and FB4 come from the same distribution.

For CT2, at .1 significance level, we conclude that the efficiencies obtained under FB1 and FB2 are lower than those obtained under FB3 and FB4.

For CT3, at .1 significance level, we conclude that the efficiencies obtained under FB2 are larger than those obtained under FB1 and FB4. At .1 significance level, we cannot reject the hypothesis that the efficiencies obtained under FB1, FB3 and FB4 come from the same distribution.

Overall, FB1 leads to the lowest efficiencies. FB2 leads to the highest efficiencies in CT3, where there is an easy threshold problem and winning levels are not restrictively high compared to the private valuations. FB3 and FB4 shine in CT2, where there are difficult threshold problems and winning levels are much higher than the private valuations. It seems the threshold problem manifests itself in CT2 cases, i.e. in cases small bidders should win but have little leeway to do so. It also manifests itself in CT3 cases, but in a diminished fashion, and is, aside from some outliers, not present in CT1 cases.

One would expect FB3 and FB4 to be most effective in cases where the winning levels are restrictively high, i.e. greater than the private values of losing bidders, and hence the information is not of much use to bidders. In such a case, FB3 and more so FB4 have the largest potential effect because of the new information they provide. Evidently, this conceivable happens more often in difficult threshold cases. In cases where the winning level is not too high for losing bidders, FB3 and FB4 do not bring a lot of extra information to the table compared to bid states feedback. In easy or insurmountable threshold cases, this is more likely to be the case.

Auction revenue results

Looking at efficiencies alone does not give a complete picture. Hence, in addition to looking at economic efficiency, we examine auction revenues, i.e. the auctioneer's surplus. Tables 3.8a — 3.8b contain the mean AS(X) data.

	STR1	STR2	STR3	STR4
FB1	0.893	0.898	0.891	0.932
FB2	0.927	0.916	0.927	0.943
FB3	0.912	0.922	0.887	0.940
FB4	0.916	0.915	0.922	0.922

(a) Mean AS(X) for factors FB and STR.

	CT1	CT2	CT3
FB1	0.929	0.917	0.848
FB2	0.943	0.945	0.882
FB3	0.920	0.942	0.863
FB4	0.933	0.945	0.858

(b) Mean AS(X) for factors FB and CT.

Table 3.8: Mean AS(X).

One first observation is that revenues in CT3 auctions are clearly lower than in auctions with an easy threshold problem. Bidders have plenty of margin to outbid the package bidder, and need not bid high compared to their private valuations in order to become winning.

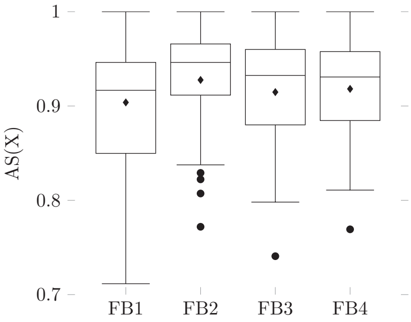


Figure 3.10: Box plots of FB.

Observation 3.5. *Auction revenues ranked by Wilcoxon-Mann-Whitney tests:*

$$FB1 \prec^* (FB2 \approx FB3 \approx FB4)$$

At .05 significance level, we conclude that the auction revenues obtained under FB1 are lower than those obtained under FB2, FB3, and FB4. At .1 significance level, we cannot reject the hypothesis that the revenues obtained under FB2, FB3, and FB4 come from the same distribution.

Efficiency and revenue progression

It is important to note that E(X) and AS(X) are naturally correlated. Looking at the measures separately does not paint a complete picture. In Figure 3.11 we depict a unique way of looking at both efficiency and revenue round-by-round progressions. That figure shows, for every level of CT (corresponding to the rows), both the average efficiency progression (left column) and the average revenue progression (right column). The horizontal axis in Figure 3.11 depicts the auction progression, e.g. if progression is at 50%, half of the total post factum required rounds have passed. This way, we can get an understanding of both average efficiency and average revenue progressions. Our observations are as follows.

Observation 3.6. *In CT1, FB3 quickly leads to higher efficiencies and revenues compared to the other FB levels, whereas FB1 performs the worst. In CT2, the progression starts similar for all FB levels, but FB3 and FB4 end up in higher efficiencies. FB1 is worse than the other FB levels. In CT3, we see that auctions with FB4, on average, start with higher efficiencies and revenues, but in the last half of the auction FB2 surpasses the other FB levels both in efficiency and auction revenue. FB1 is again the worst.*

3.7.3 Bid prices and bidder surplus BS(X)

Table 3.9 presents the average ratio of the bid prices to their private valuations over all bids expressed in the auctions. In CT1 we see that with more feedback, bidders on average bid a higher percentage of their private valuations. Basically, the package bidder has a harder time, because the competition with the smaller bidders - even though they cannot win - is stronger, leading to the package bidder having to bid higher prices, as is reflected in the higher auctioneer revenue results. In CT2 cases, we see that when FB3-4 is given, bidders on average bid higher compared to FB1-2 cases. The coalitional feedback appears to convince bidders to bid higher, which, as discussed in the previous section, leads to higher efficiencies. In CT3 cases, when FB3-4 is given, average bid prices are lower. This makes sense, as there is more room for smaller bidders to overcome the threshold. Bidders do not overshoot their bid prices to win, but can rely on the coalitional feedback to bid smarter (i.e. win at lower prices).

Table 3.10 contains the percentage of available surplus that goes to the bidders, i.e. $\frac{BS(X)}{V(X^E)}$. We can see that the total bidder's surplus in the CT1 and CT3 cases is higher for FB3-4 cases compared to FB2. This indicates that bidders make good use of the feedback in the sense that their bids do not 'overshoot' deadness levels significantly.

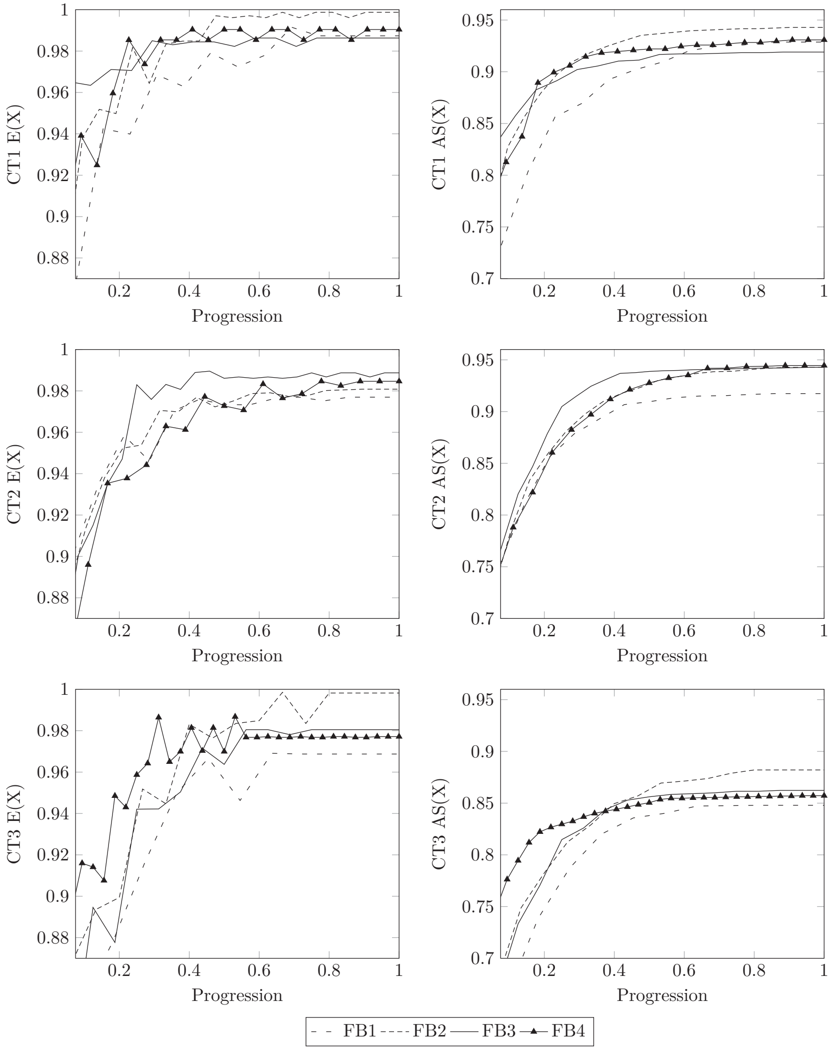


Figure 3.11: Average E(X) (left column) and AS(X) (right column) progression, per level of CT.

	CT1	CT2	CT3
FB1	0.812	0.839	0.833
FB2	0.842	0.837	0.820
FB3	0.839	0.850	0.816
FB4	0.821	0.853	0.803

Table 3.9: Average bid as percentage of private valuations.

We see that in the CT2 case, if FB1 is applied the highest bidder profits are obtained: this is an indication of the threshold problem. When more feedback is given in CT2 cases, total bidder profits lower, yet efficiencies rise as bidders are better able to coordinate and compete with the large package bidder. The bidder surplus in the FB1-CT2 case is rather high, but can in fact be explained by the threshold problem: in cases where the efficiency is not 100%, the large package bidders receive the total bidder surplus while it is not economically efficient for them to receive any surplus at all.

	CT1	CT2	CT3
FB1	0.057	0.060	0.121
FB2	0.056	0.038	0.116
FB3	0.067	0.046	0.118
FB4	0.060	0.040	0.120

Table 3.10: Average percentage of surplus obtained by the bidders $(\frac{BS(X)}{V(X^E)})$.

3.7.4 Cognitive limits

Figure 3.12 depicts the average number of different packages a bidder bids on per round. This excludes the bids entered in the first round, as bidders are then still discovering their private valuations and enter a lot of bids.

This result confirms the findings of Kagel et al. (2010) and

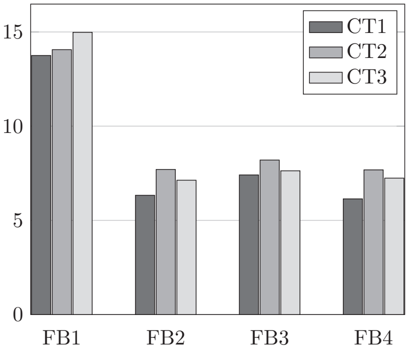


Figure 3.12: Average number bids entered per bidder per round (excluding round 1) on unique packages.

Scheffel et al. (2012), who observe that bidders usually bid on a limited number of different packages, independent of the auction format.

Observation 3.7. *Bidder support in the form of feedback reduces the number of packages bidders bid on in an auction. Bidders can focus on fewer packages, and still achieve higher efficiencies and revenues compared to basic outcome feedback.*

3.7.5 Auction Duration

Tables 3.11a—3.11b contain the mean number of auction rounds. Auctions with FB2-3-4 seem to last a little longer on average than auctions with FB1. When examining the duration in relation with the threshold and feedback factors, we observe that in the CT2-3 cases, the number of rounds is higher compared to the CT1 case, for all levels of feedback. It appears that when the threshold problem manifests itself, bidders require more rounds to reach the final auction outcome. However, in the FB1 cases, where not a lot of coordination is possible between bidders, the number of auction rounds is lower, compared to the FB2-3-4 cases.

	STR1	STR2	STR3	STR4
FB1	6.8	8.1	6.2	6.7
FB2	8.5	9.2	8.1	9.8
FB3	7.6	7.8	10.6	8.4
FB4	10.3	8.7	11.5	7.8

(a) Mean number of rounds by factors FB and STR.

	CT1	CT2	CT3
FB1	6.4	7.3	7.1
FB2	8.5	9.3	8.7
FB3	6.7	9.8	9.0
FB4	6.8	9.8	13.0

(b) Mean number of rounds by factors FB and CT.

Table 3.11: Mean number of rounds.

We test the differences in number of rounds it takes to close an auction.

Observation 3.8. *Number of rounds ranked by Wilcoxon-Mann-Whitney tests:*

$$\mathit{STR1} \approx \mathit{STR2} \approx \mathit{STR3} \approx \mathit{STR4}$$

$$\mathit{CT1} \prec^{**} (\mathit{CT2} \approx \mathit{CT3})$$

$$\mathit{FB1} \prec^{*} (\mathit{FB2} \approx \mathit{FB3} \approx \mathit{FB4})$$

At .1 significance level we cannot reject the hypothesis that the number of rounds obtained under the various structures come from the same distribution. However, at .01 significance level, we conclude that auctions where there are coordination or threshold challenges (CT2-3), require more rounds than CT-1 auctions. We also find that auctions with FB1 take fewer rounds compared to auctions with FB2, FB3, and FB4.

In addition to the number of rounds, we examine the average round duration (in seconds) in Tables 3.12a-3.12b.

	STR1	STR2	STR3	STR4
FB1	122	130	197	179
FB2	111	128	166	169
FB3	113	175	150	165
FB4	114	133	157	175

(a) Mean auction duration (seconds) per round for factors FB and STR.

	CT1	CT2	CT3
FB1	154	168	143
FB2	138	151	139
FB3	184	136	131
FB4	170	149	105

(b) Mean auction duration (seconds) per round for factors FB and CT.

Table 3.12: Mean number of seconds per round.

The results of Wilcoxon-Mann-Whitney tests are as follows.

Observation 3.9. *Round duration (in seconds) ranked by Wilcoxon-Mann-Whitney tests:*

$$STR1 \prec^{*} STR2 \prec^{**} (STR3 \approx STR4)$$

$$CT3 \prec^{*} (CT1 \approx CT2)$$

$$FB1 \approx FB2 \approx FB3 \approx FB4$$

We conclude that as the number of items and bidders increases, auction rounds tend to take more time. Additionally, in cases where there are easy threshold problems, auction rounds take less time. Overall, feedback does not seem to increase the round duration.

3.8 Conclusions

In situations where bidders have different additive or super-additive private valuations, academic literature has shown that combinatorial auctions have the edge over (sequential) single item auctions. However, combinatorial auctions introduce two problems: the coordination problem and the threshold problem. The coordination problem arises when bidders fail to identify bids that are individually profitable and collectively complementary. The threshold problem represents the next problem: even when individually profitable and collectively complementary packages are identified, and the coordination problem is essentially overcome, the problem of determining bid prices still remains. This is in fact complicated by free-rider incentives. Naturally, these problems are of significant practical interest.

In this chapter, we propose quantitative measures for both the coordination problem and the threshold problem. Our measure for the coordination problem is independent of auction formats and bids, and can be calculated using information about the private valuations of bidders. It can be used to compare different valuation designs and determine which entails the greatest coordination challenge. Our measure for the threshold problem takes a snapshot of an auction, and looks at the margin for price increases a losing coalition of bids still has compared to the current winning one. Additionally, these measures can be used to design interesting sets of private valuations for further experiments.

Keeping in mind the potential impact feedback can have in reducing the complexity for bidders to understand what is going on in a combinatorial auction, we also design new types of feedback dubbed coalitional feedback, which are specifically designed to help overcome coordination and threshold problems. With feedback that can assist bidders in considering what packages to bid on, and at what prices, bidders can bid more effectively. This is an advantage both for the auctioneer, who prefers an efficient outcome with high revenue, and the bidders, who prefer to win when they can, rather than lose because of a lack of information, coordination, or even understanding of the intricacies of combinatorial bidding.

Finally, we put different types of feedback to the test in a laboratory setting with human bidders, using iterative combinatorial auctions. These have the advantage of being an intuitive generalization of the traditional English auctions. In line with Adomavicius et al. (2012), we find that bid state feedback is a big improvement upon outcome feedback, both with respect to economic efficiency and auction revenue. Building on that result, we find that coalitional feedback can improve upon bid states feedback, leading to higher efficiencies when threshold problems are difficult. Additionally, we find that in difficult threshold cases, bidders are not insensitive to bid price suggestions and tend to follow such suggestions readily. This is interesting, as it appears that the free-rider aspect is at least diminished by coalitional feedback. The closing rule for ascending combinatorial auctions then becomes crucial, as bidders need enough time to react to coalitional feedback. In easy threshold cases, bidders also benefit from coalitional feedback, as it allows bidders to bid in a smarter fashion; they have a better idea of interesting bid prices between the deadness and winning levels. This leads to high efficiencies, but also to higher bidder profits compared to bid states feedback.

The results regarding factual and coalitional feedback are in line with Bichler et al. (2017), who find that coalitional winning levels - implemented as a pricing rule - lead to better efficiencies in their threshold value model. In both their mix value model and symmetry value model, they find little difference. However, in our experiments bid states feedback seems to lead to overall better efficiencies than Bichler et al. (2017) report, making the distinction between bid states feedback on the one hand and coalitional feedback on the other hand more difficult in our experiments. There are a number of possible explanations for this lack of clear distinction, which all deserve and require further research. One explanation is that in our experiments, only few situations occur where bid states feedback and coalitional feedback could make a difference, e.g. where the winning level is restrictively high above the private valuations of bidders while the coalitional feedback suggestion is obtainable. Another explanation is that perhaps the price suggestions are not followed enough in the rounds where it mattered, i.e. those where a coalition of small bidders is not winning but could have won together, but not all coalition members follow the suggestions. The question is then how many bidders need to follow the price suggestions in order for there to be an impact efficiency-wise. Yet another explanation could be that the coordination and/or threshold problems present in our experiments are not difficult enough for a difference between bid states feedback and coalitional feedback to be able to manifest itself.

In order to increase the penetration of combinatorial auctions in (online) markets, further exploring coalitional feedback as a bidder tool is a promising avenue of research, as it both reduces the complexity bidders face with package bidding, and allows them to focus more on relevant packages as well as adjust their bid prices smartly, considering both the coalitional winning levels as well as a number of coalitions.

Chapter 4

Conference Scheduling - A Personalized Approach

> The key is not to prioritize what's on your schedule, but to schedule your priorities.
>
> *Stephen Covey*

4.1 Introduction

Scientific conferences have become an essential aspect of (academic) life. They allow researchers (i) to present their work and receive feedback, (ii) to learn from attending talks, poster sessions, or discussion panels, and (iii) to meet with colleagues, thereby inducing new collaborations. However, attending a conferences requires a considerable effort in terms of time (e.g. preparing talks, traveling time) and money (e.g. registration fees, traveling expenses, hotels) from their participants. Conferences also have a non-negligible

This chapter is based on Vangerven et al. (2017a).

environmental impact (Gremillet, 2008). In fact, there is some debate about the value of scientific conferences, see e.g. Grant (2014), and how to lessen the carbon footprint of a conference ((Nathans and Sterling, 2016)). Obtaining exact figures with respect to the amount of money involved in organizing scientific conferences seems difficult; it is written in Ioannidis (2012) that "*an estimate of more than* 100.000 *medical meetings per year may not be unrealistic . . . the cumulative cost of these events worldwide is not possible to fathom*". Note that this figure applies to medical conferences alone.

Given these considerations and investments, it is the responsibility of the organizers to maximize the value of a conference as much as possible. Here we focus on the construction of a conference schedule that allows participants to maximally benefit from participating. Or, making this even more concrete, the schedule should enable participants to attend the talks of their interest. This clearly benefits speakers as well, potentially increasing both the size and the level of interest of their audience. Typically, a conference schedule groups talks into *sessions* (a set of talks taking place consecutively in the same room); consecutive sessions are separated by a break. Furthermore, the vast majority of conferences feature several sessions taking place at the same moment in time, i.e. sessions are scheduled *in parallel*. Consequently, a participant may be confronted with times where several attractive talks compete for his/her attendance (i.e. a *scheduling conflict*), while at other times (s)he finds nothing of interest in the schedule. A small example is given in Figure 4.1, which depicts two alternative conference schedules. In schedule 1, the participant needs to choose between preferred talks A/C, and J/L. In other words: that participant can only see half the talks he or she actually wants to see. This is not the case in schedule 2.

One popular approach to schedule conferences is *track segmentation* Sampson (2004). The organizer groups talks that cover a similar topic or method into tracks or clusters, which are then assigned to a room and scheduled in parallel. Note that a track can consist of multiple sessions. If a participant were only interested in talks from a single track, then (s)he can stay in that track's room for

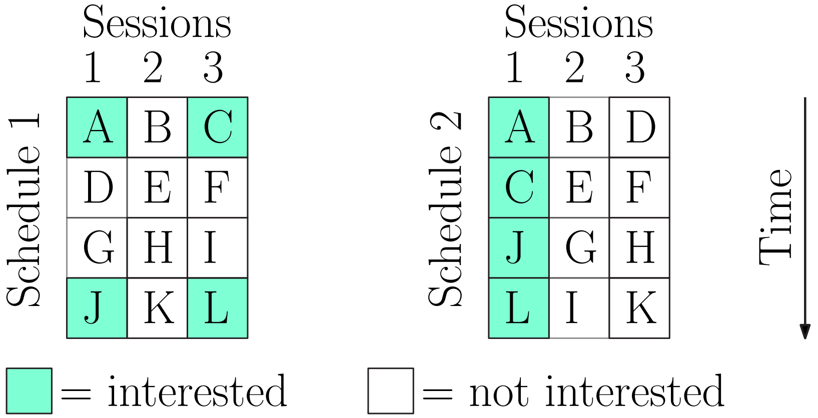


Figure 4.1: The impact of the schedule on attendance.

the duration of the conference without experiencing any scheduling conflict. However, apart from difficulties in forming meaningful clusters, track segmentation is not very effective if the participant's preferences are diverse, and not restricted to one particular topic.

In this work, a participant is expected to provide a list of preferred talks, which he or she would like to attend. Our goal is to develop a conference schedule that maximizes the participants' satisfaction. Primarily, this means we want to avoid scheduling conflicts, thereby maximizing total attendance. Next, as a secondary goal, we want to minimize *session hopping*. Indeed, confronted with multiple talks of interest scheduled in different sessions, a participant is forced to move between several sessions in order to attend as many of his or her preferred talks as possible. We call this phenomenon session hopping, and its presence is a clear indication of the existence of strong preferences of participants. Session hopping can be perceived as disturbing by presenters and their audiences. Moreover, the session hopper still tends to miss parts of the preferred talks, due to the time it takes to switch rooms and presenters not always starting at exactly the scheduled time. Finally, motivated by practical considerations, we also take presenter availabilities into account.

Our main contribution is the description of a method for the planning of a (scientific) conference. Based on given preferences of the participants, our method schedules individual talks in order to maximize total attendance; this is in contrast to many other approaches that work on the level of sessions or streams. As a secondary, original criterion, we take session hopping into account, aiming for schedules that allow participants to stay within the same room during a session. We are the first to incorporate session hopping in our scheduling approach, as session hopping is either assumed to be forbidden or non-existing in the literature, as opposed to regular participant practice. Our method has been used to schedule four scientific conferences, namely MathSport 2013, MAPSP 2015, MAPSP2017 and ORBEL2017 — we give a detailed account of our experience with the method.

We provide an overview of related work in Section 4.2. A detailed problem definition, is given in Section 4.3, followed by computational complexity results in Section 4.4. Next, we describe our solution method in Section 4.5. Finally, we present case studies on the MathSport 2013, MAPSP 2015, MAPSP2017 and ORBEL 2017 conference in Section 4.6. We finish with conclusions in Section 4.7.

4.2 Literature review

Thompson (2002) discerns two approaches to conference scheduling: a *presenter-based perspective* (PBP) and an *attender-based perspective* (ABP). With a PBP, the main goal is to meet time preferences and availability restrictions of the presenters. On the other hand, from an ABP, participants' preferences are solicited, in order to maximize their satisfaction. In the rest of this section, we will first discuss contributions that focus on the PBP, continue with papers that follow an ABP, and conclude with a few papers that solve subproblems of conference scheduling. Although we focus here on scheduling scientific conferences, there is also literature on scheduling meetings that are based on preferences of the participants; we mention Yingping et al. (2012), Ernst et al. (2003) and Ernst et al.

(2008).

4.2.1 Presenter-based perspective

Potthoff and Munger (2003) discuss a problem where sessions need to be assigned to time periods (rooms are ignored). The authors assume that the clustering of talks into sessions has already been done, in a way that each session belongs to a subject area. The goal is to find a schedule that spreads the sessions for each subject area among the time slots as evenly as possible, ensuring that no presenter has other duties (e.g. being discussant) in simultaneous sessions. An IP formulation is presented and applied to a problem instance extracted from a past meeting of the Public Choice Society, including 96 sessions and over 300 participants. This problem is revisited by Potthoff and Brams (2007), who extend the IP formulation to take into account presenter availabilities. Furthermore, their method is applied to schedule two Public Choice Society meetings, with 76 and 45 sessions.

Edis and Sancar Edis (2013) consider a very similar problem, but at the level of talks instead of sessions. Each talk has a given topic, and should be assigned to a session and a time period, such that all talks in each session have the same topic, and the occurrence of simultaneous sessions with the same topic is minimized. Furthermore, the number of talks in different sessions with same topic should be balanced, and some talks cannot be scheduled simultaneously. The authors also discuss an extended setting where presenters have preferred and non-preferred days. An IP formulation is presented, which is used to solve a hypothetical instance, including 170 talks on one of 10 topics, to be scheduled into sessions of at most 5 talks, over 12 time periods.

Nicholls (2007), like Potthoff and Munger (2003), also assumes that papers have been assigned to sessions beforehand by the organizers, but includes room assignment. The problem at hand is to assign each session to a room and a time period, such that no presenter is scheduled at two sessions simultaneously. The goal is to maximize the number of presenter preferences (e.g. preferred day or time slot) met. Participant preferences are not elicited, but can be included implicitly by the program chair, for instance by allocating appropriate rooms to sessions based on expectations regarding attendance. The author presents an algorithm, which is essentially a step-wise constructive heuristic, complemented with a set of rules to accommodate preferences and resolve conflicts. Nicholls (2007) applied his method to schedule a Western Decision Sciences Institute annual conference. This conference had over 300 participants, involving over 80 sessions and spanning 4 days.

4.2.2 Attender-based perspective

An early attempt to optimize participant satisfaction is by Eglese and Rand (1987), who collect a list of 4 preferred sessions (and one reserve session) from each participant. In their conference scheduling problem, sessions need to be assigned to time periods and rooms such that the sum of the weighted violations of session preferences is minimized. Furthermore, sessions can be offered multiple times, a decision which is also part of the problem. Although the number of rooms is limited and some rooms are not equipped with the right facilities for some sessions, room capacity is assumed to be always sufficient. The paper reports the scheduling of the national Tear Fund conference, including 15 distinct sessions, over 4 time periods and 7 rooms. As an IP formulation for a problem of this size was deemed intractable at the time, the problem was solved using simulated annealing.

Sampson and Weiss (1995) extend the Eglese and Rand (1987) setting as they consider rooms with finite seating capacities. They present a heuristic procedure that simultaneously assigns session offerings to time periods and rooms, and decides for each participant which sessions to attend (assuming that session hopping is forbidden). The procedure is tested on a number of randomly generated problem instances. Sampson (2004) describes how an annual meeting of the Decision Sciences Institute with 213 sessions to be scheduled over 10 time slots was handled using this method. Nearly half of the 1086 registered participants submitted ranked preferences for talks, which was used to rank the sessions. A post-conference survey revealed that about one quarter of the participants found the resulting schedule “much better” than in previous meetings. The method is also a part of a simulation to numerically address other issues that might be faced by a conference organizer. For instance, Sampson and Weiss (1996) discuss tradeoffs between the length of the conference, the number of offerings per session and participant satisfaction. They also investigate how seating capacity, room availability, and the utilization of time slots impact participant satisfaction.

Gulati and Sengupta (2004) enhance the problem description by Sampson and Weiss (1995) by augmenting the objective function with a prediction of the popularity of a talk, based on reviewers' assessments of the submissions and linked with time slot preferences of participants (e.g. late and last-day time slots are often poorly attended). The overall goal is to maximize the total session attendance. Gulati and Sengupta (2004) develop a solution method called TRACS (TRActable Conference Scheduling), which is essentially a greedy algorithm; no empirical results or computational analysis are reported.

The conference scheduling problem discussed by Thompson (2002) is also similar to that of Sampson and Weiss (1995). However, in Thompson (2002), meeting rooms may have different capacities, and may not always be available. He presents a method that employs a constructive heuristic followed by a simulated annealing procedure. The author performs a number of computational experiments, based on randomly generated data as well as data from a real, yet unspecified, conference. The latter includes 47 distinct sessions (some of which were to be offered 2 or 3 times), 8 time slots, and 8 rooms with different capacities. Presenters present in 1 to 5 sessions and each of the 175 participants have provided between 0 and 8 preferred sessions (neither ranked nor weighted). The author finds that his heuristic outperforms randomly as well as manually generated schedules.

Le Page (1996) assumes that each participant provides a list with a given number of sessions he or she wishes to attend. This allows to create a *conflict matrix*, where each matrix element $c_{i,j}$ represents the number of participants that wish to attend both sessions i and j. The problem is to assign the sessions to time slots and rooms (with different capacities), such that the sum of conflicts between simultaneous sessions is minimized. Furthermore, sessions with the same topic must be assigned to the same room, and some sessions need to be planned consecutively on the same day. The author develops a semi-automated heuristic in four steps, which is used to schedule a meeting of the American Crystallographic Association. This meeting includes 35 sessions, to be assigned to 5 rooms and 7 time periods. Months before the conference, preferences were solicited from the 1100 participants; about 10% of them provided a list of 7 preferred sessions. Most popularity predictions based on this input turned out to be accurate during the actual conference.

Ibrahim et al. (2008) focus on a conference scheduling problem where talks need to be assigned to time slots (spread over a number of days) in 3 parallel tracks. Each talk belongs to a field, and the schedule should be such that talks of the same field do not occur simultaneously. Furthermore, it should be avoided to schedule talks belonging to the same pair of fields in parallel more than once on the same day. The authors discuss construction methods, based on results from combinatorial design theory, for 3 cases. One case is based on data from the National Conference in Decision Science and includes 73 sessions, belonging to 8 fields, to be scheduled over 26 time slots and 2 days. Note that this setting does not involve grouping talks into sessions. Moreover, the sequence of the talks within a track on one day is of no importance, and all talks from the same field can be swapped without changing the solution quality.

In the so-called *preference conference optimization problem* (PCOP) as defined by Quesnelle and Steffy (2015), talks need to be assigned to a time slot and a room, such that scheduling conflicts are minimized. Furthermore, room and presenter availabilities need to be taken into account, including the fact that some presenters are involved in more than one talk and must be able to attend each one of them. Some talks are required to be offered multiple times. Quesnelle and Steffy (2015) show that PCOP is NP-hard and discuss an IP formulation, together with a number of performance considerations such as symmetry reduction. They apply their method on a problem instance, based on a PenguiCon conference with 253 talks. As no individual participant preferences were available, the authors have randomly generated this data from historical attendance data, for various choices of the standard deviation of the number of preferred talks per participant. Notice that the issue of grouping talks into sessions is not included in this problem, in fact, as in Ibrahim et al. (2008), each talk could be seen as a session.

4.2.3 Related problems

The problem of grouping talks into coherent sessions, given one or more keywords for each talk, is discussed by Tanaka et al. (2002) and Tanaka and Mori (2002). The objective function is a non-linear utility function of common keywords, with the underlying idea that papers in the same session have as many common keywords as possible, provided that the number of talks is balanced over the sessions. This problem is tackled using Kohonen's self-organizing maps Tanaka et al. (2002) and a hybrid grouping genetic algorithm Tanaka and Mori (2002). Both methods are tested on data from a conference of the Institute of Systems, Control and Information Engineers in Japan with 313 papers and 86 keywords.

Zulkipli et al. (2013) ignore session coherence as they attempt to group talks into equally popular sessions. The underlying idea is that in a setting with rooms of similar size and assuming that session hopping is forbidden, this will maximize participants' satisfaction in terms of seating capacity. Given a weight for each talk, based on preferences from the participants, the goal is to assign talks to sessions, such that the sum of the talk weights is balanced over the sessions. The authors present a goal programming method, which is applied to one case, involving 60 talks to be grouped into 15 sessions.

Martin (2005) elaborates on the sessions selection problem for the participant, given the conference schedule. He develops a decision support system for participants to determine their itinerary. Using a web-based approach, keyword preferences are elicited and matched with keywords supplied by talks, in order to produce an aggregate rating for each talk. This approach, which does not involve an optimization algorithm, has been used for a conference of the UK Academy of Information Systems. About one third of the 118 participants made use of the decision support system, however, the author was not able to predict session attendance based on the keyword ratings.

4.3 Problem Description

There are a number of crucial ingredients in our problem. First, there is a set of talks that needs to be scheduled; the set of talks is denoted by X. Second there is a set of timeslots, denoted by T; a timeslot refers to a period in time during which a number of talks are held in parallel — we assume that the number of talks that are held in parallel (i.e. the number of parallel sessions) is given and we denote that number by n. Further, we assume without loss of generality that the number of talks $|X|$ is a multiple of n (if necessary, this can be achieved by adding dummy talks) – notice that this means that $|T| = \frac{|X|}{n}$. A final ingredient of our problem are the participants, denoted by the set P, and their *profiles*.

Definition 4.1. *A profile of a participant $p \in P$ is represented by a binary vector $q(p)$ where $q(p)_i$ equals 1 if and only if participant p wishes to attend talk $i \in X$. A profile consisting of only 0 entries is called a trivial profile.*

In other words, a profile represents the preferences of a participant. All these ingredients allow us to formally state our problem. We assume that talks that are held in parallel cannot both be attended by the same participant, and that talks that are assigned to distinct timeslots can be attended by the same participant. The *attendance* of a talk denotes the number of participants attending a talk, and total attendance refers to the summed attendances over the talks. We assume that a participant, when a preferred talk is scheduled, will attend this talk, and in case multiple preferred talks are presented at the same time, (s)he will arbitrarily choose one of these talks to attend. Finally, we assume all participants attend the entirety of the conference.

Definition 4.2. *Given the participants' profiles $q(p)$, and given the number of parallel sessions n, the* Conference Scheduling Problem *with n parallel sessions (CSP-n) seeks to assign every talk to a timeslot such that each timeslot receives exactly n talks, while maximizing total attendance.*

The profiles allow us to compute the parameter $v_i := \sum_{p \in P} q(p)_i$: the number of participants who wish to attend talk i. Clearly, total attendance is upper bounded by $\sum_{i \in X} v_i$; notice that this number can be realized in case there are no parallel sessions, i.e. when $n = 1$.

From a computational complexity standpoint, CSP-n is an NP-hard optimization problem, already in case of three parallel sessions. We address this issue in more detail in Section 4.4.

It is clear that there will be several optimal solutions to CSP-n, as the grouping of the parallel talks into sessions and their order within a session do not impact attendance. Hence, as a secondary goal, we aim to minimize the total number of session hops. To describe this problem more precisely, let the length of a session be the number of consecutive talks in the session. Typically, a session consists of k consecutive talks, with $k \in \{2, 3, 4\}$. We use K to denote the set of session lengths present in the conference. We assume that sessions running in parallel must have the same length, and we call a set of n parallel sessions of length $k \in K$, a k-block. The *format* of a conference specifies, for each relevant session length $k \in K$, the number of k-blocks in the conference; this number is referred to as r_k. Given the format of the conference, we settle the composition of the sessions, taking into account the talks that are to be scheduled in parallel in order to maximize attendance.

However, the resulting schedule still leaves room to decide during which timeslots these parallel sessions are scheduled. This gives freedom to accommodate potential restrictions on the availabilities of speakers. Thus, in a final phase, we assign the k-blocks to timeslots, minimizing the number of violated presenter availabilities.

Concluding, the resulting conference scheduling problem has three objectives: maximizing attendance (4.5.1), minimizing session hopping (4.5.2), and satisfying presenter availabilities (4.5.3), which are considered hierarchically in this order. Notice that we have not taken room capacities into account; in Section 4.6 we describe how, if necessary, this issue can still be dealt with.

4.4 Computational complexity of CSP-n

In the following two theorems, we respectively show that the CSP-2 is polynomially solvable, and that the CSP-n is NP-hard for $n \geq 3$.

Theorem 4.1. *CSP-2 is solvable in polynomial time.*

Proof. We show that CSP-2 is a special case of the minimum weight perfect matching problem, which is polynomially solvable Lovász and Plummer (1986). Given a graph $G = (V, E)$, a matching in G is a set of pairwise non-adjacent edges. A perfect matching is a matching which matches all nodes of G, i.e., every node is incident to exactly one edge of the matching.

The reduction goes as follows. Given an instance of CSP-2, we construct a complete, edge-weighted, graph $G = (V, E)$ such that each talk in CSP-2 corresponds to exactly one node in G; thus $V := X$. For every distinct pair of talks i and $j \in X$, we calculate a coefficient $c_{i,j}$ capturing how much attendance is missed if both talks i and j are planned simultaneously, i.e., we set $c_{i,j} := |\{p \in P : q(p)_i = q(p)_j = 1\}|$. Since (i) any solution to CSP-2 can be regarded as $\frac{|X|}{2}$ pairs of talks, and hence is a perfect matching, and (ii) the coefficient $c_{i,j}$ equals the missed attendance when talks i and j are planned simultaneously, the result follows. □

Observe that while the proof of Theorem 4.1 shows that CSP-2 is a special case of minimum weight perfect matching, the converse is true as well. Indeed, when we interpret, in a given instance of minimum weight perfect matching specified by a graph $H = (W, F)$, each node in W as a talk, and the cost associated to an edge $e = (i, j) \in F$ as the number of participants wishing to see both talks i and j, it is not difficult to see that a matching with a certain cost corresponds to a schedule with that cost as the missed attendance. All this essentially implies that CSP-2 cannot be solved faster than minimum weight perfect matching.

Theorem 4.2. *CSP-n is NP-hard for each fixed $n \geq 3$.*

Proof. We will prove that the decision variant of CSP-3, which asks the question "Given a number of talks and a set of participants with corresponding profiles, does a schedule consisting of 3 parallel sessions exist such that no attendance is missed?", is NP-complete. To prove this, we will use the *Triangle Partition Problem* (TPP), which is known to be NP-complete even for graphs with a maximal degree of at most four van Rooij et al. (2012). An instance of the TPP is a graph $G = (V, E)$ with $|V| = 3\ell$, where $\ell \in \mathbb{N}_0^+$. A triangle is a collection of three nodes in G such that each pair is connected by an edge. The question that TPP asks is then: "Can the nodes of G be partitioned into ℓ disjoint sets $V_1, V_2, \ldots, V_\ell$ each containing exactly 3 nodes, such that each of these V_i is the node set of a triangle in G?".

We transform an arbitrary instance of TPP into an instance of the CSP-3. Each node in G will correspond to a talk in CSP-3, i.e. $X := V$. We set $P := \{(i, j) : i, j \in V,\ i \neq j,\ (i, j) \notin E\}$. Next, for all $p \in P$, say $p = (i, j)$, we have

$$q(p)_x = \begin{cases} 1 & \text{if } x = i \text{ or } x = j, \\ 0 & \text{otherwise.} \end{cases}$$

Informally, for every non-existing edge in the instance of TPP, we have a participant in CSP-3 who wishes to see the corresponding two talks. This completely specifies an instance of CSP-3.

Suppose the instance of TPP is a yes-instance. Then, by definition, ℓ disjoint triangles must exist in G. The three vertices of each triangle correspond to three talks that we schedule simultaneously, i.e., in parallel. Next, the parallel talks are assigned to timeslots in any order. Note that no participant misses a talk, since no participant exists that wishes to see two (or more) talks from the triple of talks scheduled in parallel. Thus, the resulting instance of CSP-3 is a yes-instance. Conversely, suppose that we have a yes-instance of the decision variant of CSP-3. Hence, we know which talks are scheduled in parallel. From this, we easily find a partition into triangles: simply select the nodes corresponding to the parallel talks as the nodes of a triangle. These nodes must correspond to triangles in G, because otherwise there would have been a missed attendance. From this it follows that CSP-n is NP-hard for $n \geq 3$. □

This complexity result is tight, in the sense that CSP-n is NP-hard even if each participant has only 2 preferred talks in his/her profile (and the problem becomes trivial if each participant has at most one preferred talk). Furthermore, our result strengthens the result that the preference conference optimization problem (PCOP) is NP-hard by Quesnelle and Steffy (2015). Indeed, their result is based on the presence of room and presenter availabilities, while in CSP-n every talk can be allocated to any timeslot.

4.5 Method

In this section we will explain a hierarchical three-phased approach to scheduling conferences. In the first phase (see Section 4.5.1), we maximize total attendance, based on the participants' profiles, i.e., we solve CSP-n. In the second phase, we seek to minimize the number of session hops, given that total attendance is maximal (see Section 4.5.2). Finally, in a third phase, we take into account presenter availabilities. We do this by minimizing the number of violated availability constraints, while fixing the total attendance and number of session hops at the levels obtained in the previous two phases (see Section 4.5.3).

4.5.1 Phase 1: maximizing total attendance

It should be obvious that maximizing total attendance is equivalent to minimizing total missed attendance. Informally, it is best to avoid scheduling talks that are on a same profile in parallel, but we still need to join talks together in a tuple.

Let H denote the set of all n-tuples consisting of distinct talks, $H \subseteq X^n$. For each $e \in H$, we set $c_e := \sum_{p \in P} \max\{0, \sum_{i \in e} q(p)_i - 1\}$. In words: the coefficient c_e denotes the total missed attendance if the talks in the n-tuple e are scheduled in parallel. Notice that the coefficients c_e are in fact a generalization of the conflict matrix used in Le Page (1996). Indeed, the conflict matrix indicates the missed attendance at the level of a session if two talks are scheduled in parallel sessions, while our coefficient does the same for any n parallel talks.

Next, we set up an integer programming model using the binary variable x_e which is 1 if and only if all talks in n-tuple e are planned in parallel.

$$\begin{aligned} \min \quad & \sum_{e \in H} c_e x_e & & \quad (4.1) \\ \text{s.t.} \quad & \sum_{e \in H : i \in e} x_e = 1 & \forall\ i \in X & \quad (4.2) \\ & x_e \in \{0, 1\} & \forall\ e \in H & \quad (4.3) \end{aligned}$$

Clearly, the objective function, Equation 4.1, minimizes missed attendance. The first set of constraints, Equation 4.2, ensures that every talk is included in exactly one n-tuple. Finally, Equation 4.3 indicates that our decision variables x_e are binary.

4.5.2 Phase 2: minimizing session hopping

Recall that an n-tuple refers to n talks that will take place in parallel. Phase 1 gives us $|T| = \frac{|X|}{n}$ such n-tuples; we use H^* to denote the set of selected n-tuples from phase 1. Here, in phase 2, our goal is to assemble these n-tuples into so-called k-blocks. Recall that, in line with the terminology introduced in Section 4.3, a k-block can be seen as an ordered set of k ordered n-tuples, thereby yielding n parallel sessions, each session consisting of k consecutive talks.

Consider a set of k n-tuples found in phase 1, say $e_1, e_2, \ldots, e_k$. Clearly, there are different ways to organize a set of k n-tuples into a k-block: one can permute the sequence of n-tuples e_1, e_2, $\ldots$, e_k, as illustrated in Figure 4.2, and one can permute the n talks within each n-tuple e_i, as illustrated in Figure 4.3. In total this gives $k!(n!)^k$ possibilities, i.e., given a set of k n-tuples, there are $k!(n!)^k$ distinct k-blocks corresponding to that set.

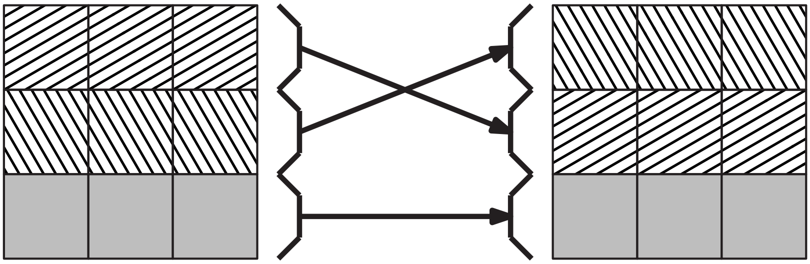


Figure 4.2: Permuting two 3-tuples (rows) in a 3-block.

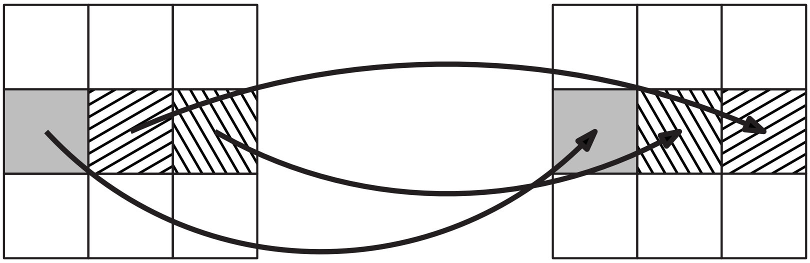


Figure 4.3: Permuting two talks in a 3-tuple.

We remark here that some permutations are equivalent with respect to session hopping. Indeed, we need not consider permutations obtained by (i) reversing the order of the n-tuples, or (ii) changing the order of the talks in the first n-tuple.

We now define the *hopping number* of a participant in a k-block as the minimum number of session hops needed by that participant to attend the maximum number of talks in this k-block (s)he is interested in. We find the *hop coefficient* of a k-block by summing the corresponding hopping numbers over all participants.

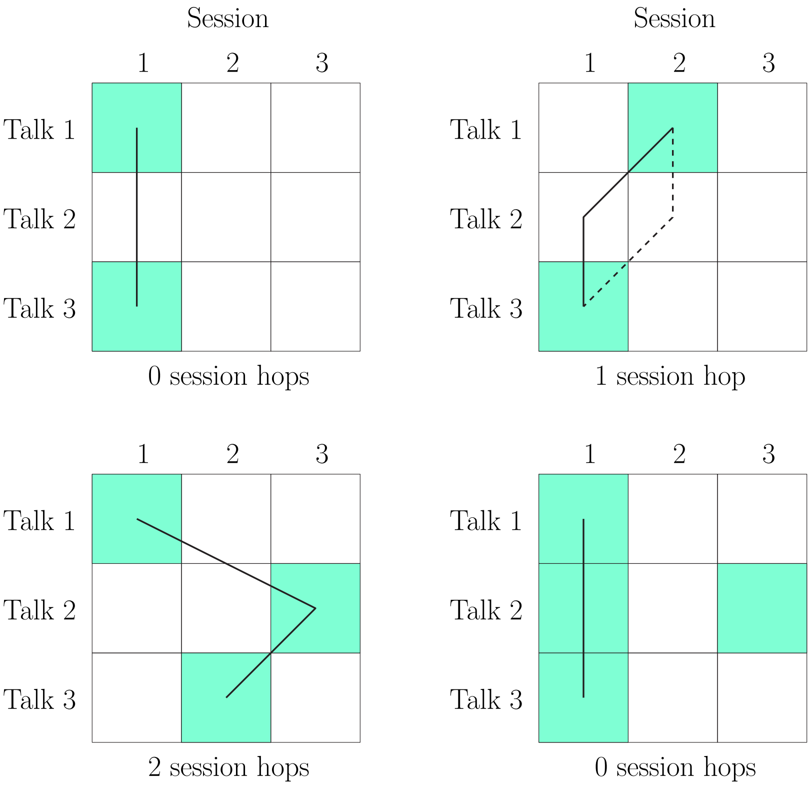


Figure 4.4: Four session hopping examples using 3-blocks.

Figure 4.4 illustrates this using a 3-block with three parallel sessions, and a number of profiles as examples. The top left example shows that the participant is interested in the first and last talk in session 1. The hopping number for this profile will therefore be 0, as indicated by the full line; the participant can stay in session 1 and not miss any of his/her preferred talks. The top right example shows the profile of a participant interested in the first talk in session 2 and the last talk in session 1. In order to attend both talks, this participant will have to switch rooms exactly once, leading to a hopping number of 1. There are two alternative ways this participant can switch, one is indicated using the full line, the other using the dashed line. Similarly, in the bottom left example, a participant with this profile will have to switch exactly twice to attend all talks of his or her interest, as indicated by the full line. The final example, on the bottom right, shows a profile and a k-block where at a particular moment in time, more than one talk of interest is planned. In that case, we assume that the participant chooses talks such that (s)he can attend the maximum number of talks of his/her interest, while minimizing the number of required session switches. In the bottom right example this means that the participant will choose to stay in session 1 for the second talk, as indicated by the full line, instead of switching to session 3 and then back to session 1.

In the following subsections, we describe two ways of actually constructing the k-blocks from the given n-tuples from phase 1: an exact approach and a heuristic approach. The exact approach refers to the fact that, given the set H^*, a solution is computed for which the sum of the hop coefficients is minimum. The heuristic approach is an approach where an approximation of the hopping number is used, and hence, no guarantee of optimality can be given. However, the heuristic approach will be very efficient computationally.

Exact dynamic programming approach

For each block consisting of n parallel sessions of length k, we can determine the minimum number of hops required by a participant p using a dynamic programming algorithm.

We define the set $N = \{1, \ldots, n\}$ as the set of parallel sessions, indexed by j, and we define the binary indicator $A(t, j)$ which equals

1 if the profile of participant p indicates that (s)he would like to attend the talk in session j during the t'th timeslot of this block, and zero otherwise. Let $H(t, j)$ be the minimum number of hops required by participant p in order to attend the talk in session j during the t'th timeslot, while also attending a talk of his/her preference during each of the timeslots $1, \ldots, t-1$ for which at least one talk is preferred by this participant. We can express the hopping number for participant p in this block b as follows:

$$H_p^b = \min_{\substack{j \in N: \\ A(x,j)=1}} H(x, j),$$

where x is the latest timeslot in the block for which this participant has at least one preferred talk. If a participant does not have a preference for any of the talks in a block b, then $H_p^b = 0$.

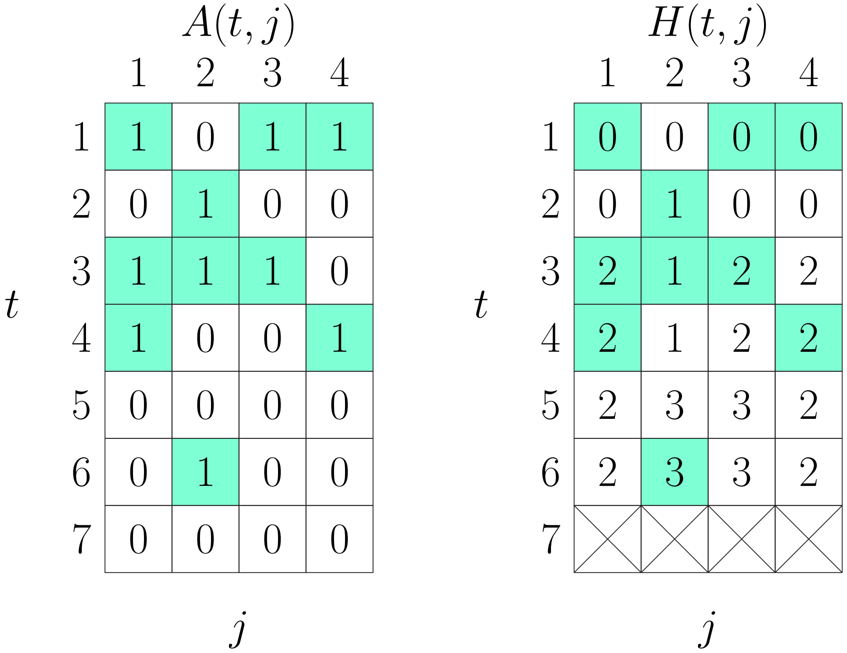
It is trivial to see that $H(1, j) = 0$ for all $j \in N$. Furthermore, it is not difficult to argue that a participant, in order to obtain a minimum number of hops up to timeslot i, should not switch from their current session (say j) to a different session (say j') unless in timeslot t (s)he has a preference for a talk in session j', while having no preference for the talk in session j. We use $\Delta_{j,j'}$ with $j, j' \in N$ as a binary indicator, which is 0 if $j' = j$ and 1 if $j' \neq j$. More formally, the following recursion holds for $i > 1$:

$$H(t, j) = \begin{cases} H(t-1, j) & \text{if } \sum_{j' \in N} A(t-1, j') = 0 \\ \min_{\substack{j' \in N: \\ A(t-1,j')=1}} (H(t-1, j') + \Delta_{j,j'}) & \text{if } \sum_{j' \in N} A(t-1, j') \neq 0 \end{cases}$$

The above implies that, for each participant p and for each block b, the hopping number can be determined using a dynamic programming algorithm. Thus, for each set of k n-tuples, we can compute a k-block b that minimizes the hop coefficient. The resulting value is denoted by w_b and represents the total number of session switches that will result from having this block b as part of the conference schedule.

We give an example that demonstrates how the recursion works for a particular participant.

Example 6. *Consider a 7-block with 4 parallel sessions. Suppose a participant has preferences $A(t, j)$ as depicted in the left of Figure 4.5. First note that the last preference occurs in the row corresponding to $t = 6$. This means that we can skip $t = 7$. Calculating the number of hops for $t = 1$ is trivial: no hops are required since this is the start of the 7-block. There are, however, three talks of interest to the participant. At $t = 2$, there is exactly one talk of interest available. Because there was at least one talk of interest at $t = 1$, we need to check whether a hop is required. Indeed; no matter which talk of interest is followed at $t = 1$, one hop is required. At $t = 3$, there are three talks of interest. Note however, that at $t = 2$ there was*



$A(t, j)$

t \ j	1	2	3	4
1	1	0	1	1
2	0	1	0	0
3	1	1	1	0
4	1	0	0	1
5	0	0	0	0
6	0	1	0	0
7	0	0	0	0

$H(t, j)$

t \ j	1	2	3	4
1	0	0	0	0
2	0	1	0	0
3	2	1	2	2
4	2	1	2	2
5	2	3	3	2
6	2	3	3	2
7	☒	☒	☒	☒

Figure 4.5: An example of the hop calculating recursion.

exactly one talk of interest, hence that will be the starting point. This results in an extra hop for the talks at $j = 1$ and $j = 3$, but not for the talk at $j = 2$. Moving on to $t = 4$, we find two talks of interest. Our starting point, however, is still $j = 2$ because it has the lowest number of hops. This results in extra hops for the talks of interest at $t = 4$. At $t = 5$ there are no talks of interest available. The lowest number of hops at $t = 4$, where there is a talk of interest, is 2. We keep this number of hops in the same column, but add a hop to all different columns. At $t = 6$, there is again one talk of interest. There was no talk of interest at $t = 5$, so we can copy the numbers of the previous row.

The resulting path for the participant that minimizes the number of required hops, while maximizing the number of talks of interest that, is now as follows. At $t = 1$, the participant can choose between the talks at $j = 1$, 2 or 3. At $t = 2$, the participant moves to $j = 2$, which is the first hop. At $t = 3$, the participant stays at $j = 2$. Note that, even though there are two other talks of interest at this time, there is no other choice available. Next, at $t = 4$, the participant moves to $j = 1$ or $j = 4$; both are equivalent with respect to the number of hops. In fact, this will be the second hop. At $t = 5$, the participant stays in the same room as $t = 4$. Finally, at $t = 6$ the participant moves to $j = 2$, which is the last hop.

We now build an integer programming model to minimize the number of session hops. We use $B(k)$ to denote the set of k-blocks ($k \in K$); recall from Section 4.3 that the format of the conference (r_k) is given. Let $\mathcal{B} = \cup_{k \in K} B(k)$, further, we write $e \in b$ to denote that block b contains e as an n-tuple. The binary variable y_b equals 1 if and only block b is included in the schedule.

$$\min \sum_{b \in \mathcal{B}} w_b y_b \tag{4.4}$$

$$\text{s.t.} \sum_{b \in B(k)} y_b = r_k \qquad \forall\ k \in K \tag{4.5}$$

$$\sum_{b \in \mathcal{B}: e \in b} y_b = 1 \qquad \forall\ e \in H^* \tag{4.6}$$

$$y_b \in \{0,1\} \qquad \forall\ b \in \mathcal{B} \tag{4.7}$$

The objective function minimizes the number of hops over all k-blocks. The first set of constraints, Equation 4.5, ensure that we select the proper number of each k-block, according to the conference format. Equation 4.6 makes certain that every n-tuple (input from phase 1) is used exactly once. The final constraint set enforces that our decision variables y_b are binary.

Heuristic Approach

We now give an informal description of a heuristic that we used to construct the required k-blocks. For ease of exposition, we assume first that we need only 2-blocks or only 4-blocks; later, we will indicate how to modify the method when k-blocks with other values of k are required as well. Recall that we are given $|T|$ n-tuples from phase 1. Let us now build a complete, undirected, edge-weighted graph $G = (V, E)$ as follows: there is a node in V for each $e \in H^*$. The cost of a pair of nodes in V that correspond to, say, $e_1, e_2 \in H^*$, is computed as follows. Consider a pair of talks i and j such that talk i is from n-tuple e_1, and talk j is from n-tuple e_2. We count the number of participants that (i) wish to attend talk i, and do not wish to attend talk j but wish to attend some other talk from n-tuple e_2, and (ii) wish to attend talk j, and do not wish to attend talk i but wish to attend some other talk from n-tuple e_1. In this way we have identified the number of participants that will incur a hop when talks i and j are scheduled consecutively in a same session - let us call this number $d_{i,j}$. We compute this number for every pair of talks, where one talk is from e_1, and the other talk is from e_2, leading to n^2 $d_{i,j}$ numbers. To compute the cost for the edge in G between nodes in V that correspond to e_1 and e_2, we solve an assignment problem based on these $d_{i,j}$ numbers. Indeed, the resulting solution value of this assignment problem is the cost of the edge in V between the n-tuples e_1 and e_2; in addition, there is an optimal assignment of talks known in case this edge from G is selected. Having built the graph G, we now compute a minimum cost perfect matching in this graph, giving us pairs of n-tuples, i.e., 2-blocks. Notice that in case the schedule would consist of 2-blocks only, the heuristic could stop here. We now proceed by applying the procedure recursively: we consider each pair of n-tuples as an entity by itself. That is, we now build a graph, say G', where each pair of n-tuples just found corresponds to a node in V'. Again, we solve an assignment problem to find the cost of selecting a pair of 2-blocks in G'. More precise, given two pairs of 2-blocks (e_1, e_2) and (e_3, e_4), there are exactly four ways in which we can concatenate them: $(e_1, e_2, e_3, e_4), (e_1, e_2, e_4, e_3), (e_2, e_1, e_3, e_4)$ and (e_2, e_1, e_4, e_3). For each of these four ways, we solve an assignment problem based on the (known) attendance in the second and third n-tuple. We keep the solution which has the smallest cost as the cost of an edge in G'. Now we compute a minimum cost perfect matching in G', giving us pairs of 2-blocks, i.e., 4-blocks, and we have found a solution.

It is clear that the cost that we base our computations on, is in fact an approximation of the true hopping coefficients. This means that the outcome of the heuristic is not necessarily leading to a solution with a minimum number of hops. Observe that the method is efficient: in order to compute all cost coefficients, we need to solve $O(\binom{|T|}{n})$ assignment problems of size $n \times n$ (where, in practice, n will be small), and two minimum cost matching problems.

Let us finally indicate how to modify the method when k-blocks for other values of k need to be found – we assume that $k \in \{2, 3, 4\}$, as is the case for all conferences we encountered. First, we run the heuristic as described. We select from the resulting solution the r_4

best 4-blocks. Next, we remove these n-tuples from the instance, and run the first step of the heuristic to find a set of 2-blocks. We select the best r_2 blocks, and remove the corresponding n-tuples from the instance. Then we add r_3 dummy n-tuples to the instance, where a dummy n-tuple corresponds to a node in V that has zero cost to each non-dummy n-tuple, and a high cost to another dummy node. Solving a minimum cost matching in G results in $2r_3$ 2-blocks, which we use to construct G'. However, we assign a high cost to edges between two 2-blocks that both have a dummy n-tuple, and of the four ways in which two pairs can be concatenated, we only consider options where the dummy n-tuple is in the first or the last position. Computing a minimum cost matching in G' results in r_3 4-blocks containing a single dummy n-tuple, which we can remove to arrive at a solution with the prescribed number of 3-blocks.

4.5.3 Phase 3: presenter availabilities.

In this phase, we assign the blocks found in Phase 2 to timeslots while minimizing the number of violated speaker availabilities. As the order of the talks within a block has been settled, this phase will assign each selected k-block, and hence each talk, to a timeslot. We define $T_S \subseteq T$ as the set of timeslots that correspond to session starting times. The number of violated availabilities if block b is assigned to timeslot $t \in T_S$ is denoted by $u_{b,t}$, and can easily be computed from known presenters availabilities. We use an assignment based integer programming formulation where $z_{b,t} = 1$ if block b is scheduled to start in timeslot $t \in T_S$, and 0 otherwise.

$$\min \sum_b \sum_t u_{b,t} z_{b,t} \tag{4.8}$$

$$\text{s.t.} \sum_b z_{b,t} = 1 \qquad \forall\, t \in T_S \tag{4.9}$$

$$\sum_t z_{b,t} = 1 \qquad \forall\, b \in \mathcal{B} \tag{4.10}$$

$$z_{b,t} \in \{0,1\} \qquad \forall\, b \in \mathcal{B}, t \in T_S \tag{4.11}$$

The objective function minimizes the total number of violated availabilities. The first set of constraints, Equation 4.9, ensures that every timeslot at which sessions start gets assigned one block. The second set of constraints, Equation 4.10, ensures that every block is assigned exactly once. Finally, we enforce our variables $z_{b,t}$ to be binary.

After phases 1, 2 and 3, we have composed the schedule for the conference. Now it is easy to see how a personalized optimal itinerary can be constructed for every participant. By constructing the k-blocks for the second phase, we already know the maximum number of preferred talks each participant can attend in every k-block, as well as how many session hops are required in order to actually attain that attendance. As a result, we can simply combine this information with the timeslots that were chosen in Phase 3 to present each participant with an individual itinerary.

4.6 Practical applications

In this section we will apply our three-phased conference scheduling approach to four practical cases: MathSport International (2013), MAPSP (2015 and 2017), and ORBEL (2017). These are medium-sized conferences with 2, 3, and 4 parallel sessions respectively. For each of these conferences, we sent an e-mail to all registered participants enquiring each participant for his/her profile. Figures 4.6 and 4.7 show boxplots of the number of preferences given per participant and the number of preferences per talk respectively. One observation is that for ORBEL, a conference with talks covering a wide variety of topics, there were fewer preferences given by participants, and fewer preferences per talk compared to MathSport and MAPSP, which are more focused on specific topics. These (anonymous) profiles are publicly available[1]. The schedule obtained by applying our method was adopted by the conference organizers in each case. Furthermore,

[1] Data can be found at the following URL: http://feb.kuleuven.be/public/u0093797/cspinstances.zip

based on the profiles, we were able to select suitable session chairs for each session. Indeed, for each session we selected chairs among the participants that had expressed an interest for a maximal number of talks in that session.

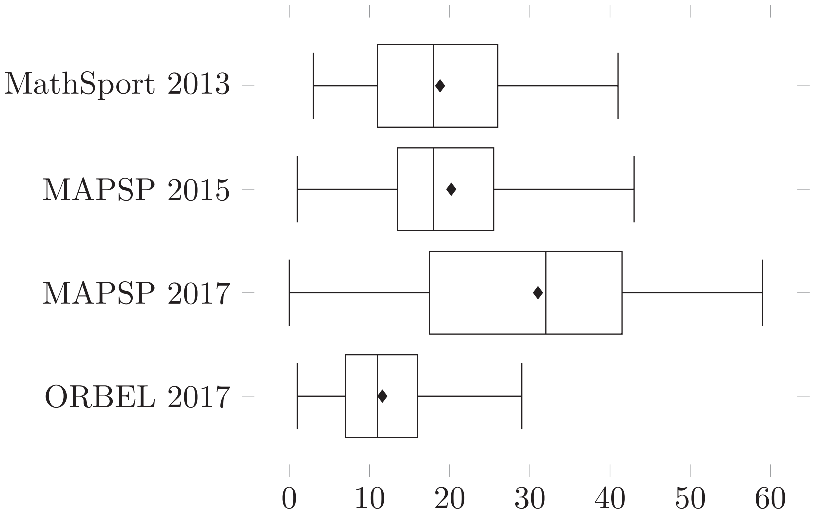


Figure 4.6: Box plots of preferences/participant.

4.6.1 MathSport 2013

MathSport International is a biennial conference dedicated to all topics where mathematics and sport meet. The fourth edition was organized in Leuven (Belgium) on June 5-7, 2013 and attracted 76 talks (apart from 3 keynote talks) and 97 participants. The conference featured two parallel sessions, and consisted of five 4-blocks and six 3-blocks.

Our preference elicitation resulted in 68 nontrivial profiles (a response rate of 70%), amounting to 1279 indicated preferences in total.

The first phase of scheduling MathSport, maximizing attendance, is an instance of CSP-2, which can be solved as a minimum weight

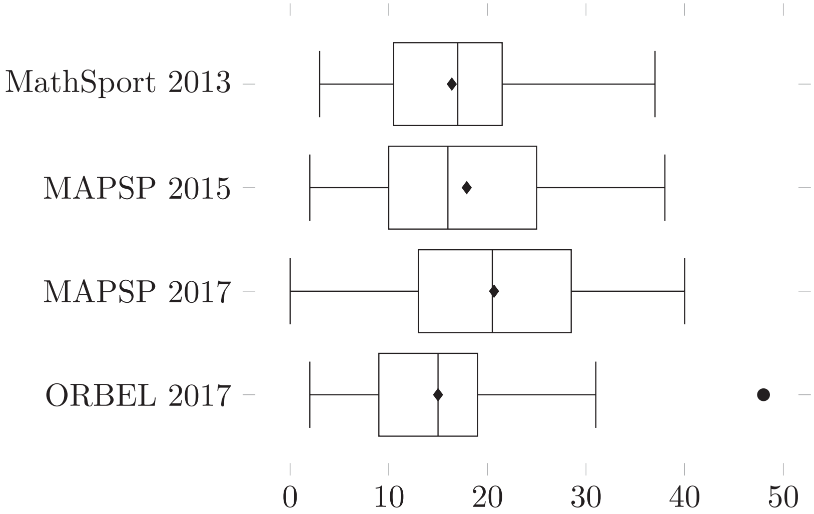


Figure 4.7: Box plots of preferences/talk.

perfect matching problem (see Theorem 4.1). For each pair of talks, the weight of an edge boils down to the number of participants that want to attend both talks. However, we had 4 speakers presenting two talks. These pairs of talks could obviously not be part of the matching, which we enforced by giving them a very high weight. We tackled this phase using a straightforward IP formulation. The optimal solution involved 42 scheduling conflicts, allowing the participants to attend 96.7% of the talks in their profile on average.

The MathSport 2013 conference was characterized by several presenter availability restrictions. Specifically, 10 talks could not be scheduled on given days, and for 2 other talks only 1 particular timeslot was acceptable. Hence, we gave priority to phase 3 over phase 2, meaning that we first grouped the pairs of talks into parallel sessions, for which the starting times were already set (using a slightly modified version of formulation (4.8)-(4.11)). The result was a timetable that did not violate any presenter unavailability.

Finally, we needed to decide for each group of paired talks to which session — the one in room A or the one in room B — the talks should be assigned, and in what order. As the capacity of both rooms was identical, minimizing session hopping was the only concern. Since there are only 8 (4) ways to organize a group of 4(3) pairs of talks into sessions, and 24 (6) different orders of these pairs within a session of 4(3) talks, this step took near-zero computation time.

4.6.2 MAPSP 2015 and MAPSP 2017

The workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP) is a biennial conference dedicated to scheduling, planning, and timetabling. The 12th edition of MAPSP was held on June 8-12 2015 in La Roche-en-Ardenne (Belgium) and the 13th edition was held on June 12-16 in Seeon-Seebruck (Germany). Both conferences featured three parallel series of sessions, spread over five days.

Specifically for MAPSP 2015, there were eight 3-blocks and three 2-blocks available for scheduling talks, leading to a total capacity for talks of 90. The MAPSP program committee accepted 88 talks, to which we added 2 dummy talks (corresponding to empty spaces in the conference schedule) in order to match the capacity. For MAPSP 2017, there were 87 talks, to be scheduled in seven 3-blocks and four 2-blocks.

We collected 78 nontrivial profiles for MAPSP 2015 and 58 for MAPSP 2017. The total number of indicated preferences were 1576 and 1799 respectively for MAPSP 2015 and 2017.

The first phase of scheduling MAPSP, maximizing attendance, is an instance of CSP-3. Remembering the coefficient c_e, as defined in Section 4.5.1, we have for each triple (3-tuple) of distinct talks $i, j, k \in X$ the number of missed attendance if talks i, j and k are scheduled in parallel: $c_{i,j,k}$. Note that this is easily computed using the profiles, as indicated in Section 4.5.1.

We used formulation (4.1)-(4.3), which for MAPSP 2015 amounts to $\binom{90}{3} = 117480$ variables and 90 constraints, and for MAPSP 2017 amounts to $\binom{87}{3} = 105995$ variables and 87 constraints. For MAPSP 2015, we obtained an optimal objective value of 155. Equivalently, the obtained triples allowed the participants to attend 1421 of the 1576 preferred talks according to the profiles. For MAPSP 2017, the optimal objective value was 478.

In the second phase of scheduling MAPSP, our goal is to assemble the triples into 3-blocks and 2-blocks such that session hopping is minimized. Recall that a 3-block, as well as a 2-block, consists of three parallel sessions, each taking place in a different room. The optimal objective value of the second phase is 120 for MAPSP 2015, which is the total number of hops for all participants. For MAPSP 2015, the minimum number of hops was 145.

Note that in order to arrive at a schedule, there is still freedom in the allocation of sessions to rooms. Indeed, the allocation of sessions to rooms does not influence the attendance or the number of session hops. This allows us to take the room capacity into account to some extent. In Section 4.5 we assumed that the rooms have infinite capacity. The available rooms at the MAPSP 2015 conference each had a different (and finite) capacity: these capacities equaled 170, 100 and 40 seats. So, with the three sessions of each 3- and 2-block known, we used the following strategy to allocate sessions to rooms. First, find in each session the talk with the largest number of votes, i.e., the talk i with maximum v_i. The session for which this number is minimal goes to the smallest capacitated room. Next, for the two remaining sessions, we sum the v_i's of the talks in each session, and put the session with the largest sum in the largest room. This system of allocation offers no hard guarantee that the room capacities are respected. However, it turned out that, using the system described above, room capacity was not an issue.

Finally, in the third phase, we need to identify a talk with a speaker and take into account the various availabilities of speakers in order to assign 2-blocks and 3-blocks to timeslots. For MAPSP 2015 a total of 13 speakers had availability restrictions (i.e. not being available on certain days), whereas MAPSP 2017 only had 4 speaker restrictions. This phase was easily handled by solving formulation (4.8)-(4.11). The solution resulted in schedules respecting all speaker availabilities, which were then implemented for MAPSP 2015 and 2017.

4.6.3 ORBEL 2017

ORBEL is the annual conference of the Belgian Operational Research Society, and serves as a meeting place for researchers working in Operational Research, Statistics, Computer Science and related fields. ORBEL 2017 took place February 2-3 2017 in Brussels (Belgium), and was the 31st edition of ORBEL. It featured four parallel series of sessions, spread over two days. Specifically, a total of one 2-block, two 3-blocks, and three 4-blocks were available to schedule talks, leading to a total capacity of 80 talks, which exactly corresponds to the number of accepted talks.

We collected 101 non-trivial profiles from 140 participants, leading to a total of 1200 indicated preferences.

The first phase of scheduling ORBEL, which maximizes attendance, corresponds to an instance of CSP-4. This leads to a total of $\binom{80}{4} = 1581580$ quadruples (4-tuples). However, not all these quadruples were feasible and hence needed to be filtered out. 30 out of 80 talks were classified as being ‘COMEX talks’. The organizing committee requested that the schedule ensures sessions consisting of only COMEX talks, such that the number of parallel COMEX sessions is minimal. Hence, we kept quadruples that had exactly 1 or 2 COMEX talks in parallel. In addition, 4 particular talks were to be grouped in an ‘ORBEL Award’ session. As they could not be scheduled in parallel, we filtered out all quadruples containing 2 or more such ORBEL Award talks. One of the jury members of the ORBEL Award also gave a talk at ORBEL, and hence could not be scheduled in parallel with ORBEL Award talks. Two presenters each had two talks, which could consequently not be in the same quadruple. Finally, 9 participants could not be present on Friday, which was the day the ORBEL Award took place. In other words:

the talks of these participants were not scheduled in parallel to the ORBEL Award talks. In the end, we ended up with 889573 feasible quadruples, a significant reduction from the 1581580 possible quadruples. We used formulation (1)-(3), and quickly found an optimal solution with objective value 100. In other words: participants could see 1100 of their 1200 preferred talks.

In the second phase of scheduling ORBEL, we assembled the 20 quadruples resulting from phase 1 into one 2-block, two 3-blocks and three 4-blocks in a way that minimizes session hopping, that ensures COMEX talks are together in 1 or 2 parallel sessions, and that groups the ORBEL Award talks in a single session. This resulted in an optimal solution of 281, the total number of hops for all participants, such that they can attend the maximum number of preferred talks. The most challenging aspect of exact approach of the second phase is undoubtedly to determine a k-block with minimal session hopping for each set of k quadruples (with $k \in \{2, 3, 4\}$), which took several hours.

Finally, in the third phase, we needed to take into account availabilities constraints in order to assign the various 2-, 3-, and 4-blocks to specific time-slots. Using an assignment-based formulation, similar to the one used for MAPSP, we found a schedule, respecting all availability constraints.

4.6.4 Computation times and heuristic results

The computation times for solving the IP formulations in phase 1 and 2 for the different conferences are presented in Table 4.1. All calculations are done using CPLEX 12.6.3 on a laptop with an Intel Core i7-4800 MP CPU @ 2.70Ghz processor and 8 GB RAM.

The results of the exact and the heuristic approach to solve phase 2 in terms of total number of hops are presented in Table 4.2. Notice that the numbers reported for the heuristic are not based on the approximated hop coefficients, but reflect the true number of hops. The quality of the heuristic is acceptable, with solution quality varying between 1 and 2.43 times the optimum. While the

	Phase 1	Phase 2
MathSport 2013	0.05	0.06
MAPSP 2015	3.89	0.12
MAPSP 2017	3.53	0.11
ORBEL 2017	61.03	0.03

Table 4.1: Computation times (in seconds).

dynamic programming approach does not scale well with the size of the conference, the required computation times for the heuristic are always 1 second or less.

	Exact	Heur	Heur/Exact
MathSport 2013	141	141	1.00
MAPSP 2015	120	292	2.43
MAPSP 2017	145	207	1.43
ORBEL 2017	281	344	1.22

Table 4.2: Hop results for exact and heuristic approach.

4.7 Conclusions

In this chapter, we argue that conference scheduling is an important and relevant problem. Indeed, conferences require significant investments (time, money) from participants, which strongly motivates a good schedule. We identify the Conference Scheduling Problem, where the goal is to maximize total attendance based on given preferences of the speakers. This problem is shown to be easy in case of two parallel sessions, and becomes NP-hard for three or more parallel sessions. The main motivation for this research however, comes from a pragmatic origin: scheduling actual conferences. We describe how we applied our three phase scheduling method to four different conferences, MathSport 2013, MAPSP 2015 & 2017 and ORBEL 2017, and discuss these cases extensively.

A possible consequence of our method could be that the resulting sessions are incoherent, since their composition is based solely on participant preferences, and not on the topic of the talk. However, as the profiles of the participants tend to contain talks on similar topics, the resulting sessions are still relatively coherent.

Another concern is that the current approach does not treat all participants equally. By preferring many talks, a participant may have a larger impact on the conference schedule than a participant with a small number of preferences. This can be remedied by giving an appropriate weight to the preferences of each participant, or limiting the number of preferences each participant can express to e.g. the number of timeslots of the conference.

A final issue is the scalability of our approach. Although our method has been developed for medium-size conferences, the question arises to what extent it scales to much larger conferences. This is relevant both when eliciting participants' preferences, as well as computationally. The latter can be accommodated by solving the second phase using our heuristic approach, as the bottleneck appears to be the computation of the hopping coefficient for all k-blocks, rather than solving the IP formulations. In case preference elicitation is unpractical due to the high amount of talks, our approach could be applied on the level of streams or tracks. Indeed, talks in large conferences are often from the beginning (i.e., when submitting an abstract) assigned to streams, and the conference schedule is typically based on track segmentation. As not all streams cover the full length of the conference, there would be possibilities to minimize overlap between pairs of streams that many participants would like to attend. Furthermore, if overlap is unavoidable, the streams could be allocated to rooms which are close to each other, facilitating session hopping. In fact, eliciting stream preferences would provide a good idea of the required room capacities for each stream.

Chapter 5

Conclusion

> We live on an island surrounded by a sea of ignorance. As our island of knowledge grows, so does the shore of our ignorance.
>
> *John Archibald Wheeler*

Combinatorial auction: theory, experiments, and practice. In this conclusion, we will go over these contributions, while highlighting possible limitations and avenues for future research.

5.1 Theory

We contributed to the theory of combinatorial auctions in chapters 2, 3 and 4. In Chapter 2, we study the winner determination problem for a combinatorial auction with a specific but relevant geometric structure. We argue that this structure is relevant, as it occurs in real estate, plots of land, mineral rights, and theaters and stadium seats. We provide computational complexity proofs for difficult cases and present a polynomial time dynamic programming algorithm; a tool that enables auctioneers to efficiently compute the winning bids in several geometrical settings. The potentially too long computing times for winner determination algorithms are often used as an argument against using combinatorial auctions (Porter et al., 2003). Hence, finding situations where such running times do not present an issue is important. The managerial implication is that (private) organisations in e.g. real-estate or governments selling licenses for oil/gas/mineral tracts might quicker consider using a combinatorial auction. However, our results may also prove useful for experimental research: our dynamic program will allow researchers to study bidder behavior in larger settings, involving more items and bidders than considered so far.

The analyses in Chapter 2 are specifically for single-minded bidders or bids entered using an OR-bids bidding language, in a specific geometrical setting. While this is a limitation of Chapter 2, it immediately presents several interesting avenues for future research here; one could look for more (practical) geometrical settings and investigate the computational complexity, or extend the dynamic programming algorithm to allow for XOR-bids/sub-additive private valuations.

In Chapter 3 we propose two quantitative measures; one that measures the coordination problem and one that measures the threshold problem. The coordination problem arises when bidders fail to coordinate their bids, and fail to identify packages that are individually profitable and collectively complementary. The threshold problem represents the next step: when the coordination problem is essentially overcome, the problem of determining bid prices still remains, and is in fact complicated by free-rider incentives. Naturally, these problems are of significant practical interest. These measures can be used to design interesting sets of private valuations that can be used in future experiments. Also in Chapter 3, we design coalitional feedback, which aims at helping bidders overcome coordination and threshold problems. With feedback that can assist bidders in tactfully considering what packages to bid on, and at what prices, bidders can bid more effectively. This is an advantage both for the auctioneer, who prefers an outcome with high revenue, and the bidders, who prefer to win when they can, rather than lose because of a lack of information, coordination, or even understanding of the intricacies of combinatorial bidding.

Finally, in Chapter 4, we identify the Conference Scheduling Problem, where the goal is to maximize total attendance based on given preferences of the speakers. This problem is shown to be easy in case of two parallel sessions, and becomes NP-hard for three or more parallel sessions.

5.2 Experiments

We contribute to experiments in Chapter 3. We put different types of feedback to the test in a laboratory setting with human bidders, and opted for the intuitive ascending combinatorial auctions. These have the advantage of being an intuitive generalization of the traditional English auctions, and have the added advantage that bidders need to enter bids on all possible packages as would be the case in most sealed-bid formats (e.g. the generalized Vickrey-Clarke-Groves combinatorial auction). In line with Adomavicius et al. (2012), we find that bid state feedback is a big improvement upon outcome feedback with respect to both economic efficiency and revenue. Building on that result, we find that coalitional feedback can improve upon bid state feedback, leading to higher efficiencies especially when threshold problems are difficult. Additionally, we find that in difficult threshold cases, bidders are not insensitive to bid price suggestions and tend to follow such suggestions readily. This is interesting, as it appears that the free-rider aspect is at least diminished by coalitional feedback. The closing rule for combinatorial auctions then becomes crucial, as bidders need enough time to react to coalitional feedback. In insurmountable threshold cases, coalitional feedback creates a more competitive environment for the dominant package bidder, leading to higher prices paid. In easy threshold cases, bidders also benefit from coalitional feedback, as it allows bidders to bid in a smarter fashion; they have a better idea of interesting bid prices between the deadness and winning levels. This leads to higher efficiencies, but also higher bidder profits.

In order to increase the penetration of combinatorial auction in (online) markets, good bidder support is important. Iterative combinatorial auctions are an attractive combinatorial auction format because they are a natural generalisation of the well-known English auctions. They are intuitive, and as such require less effort for bidders to understand. In addition, iterative combinatorial auctions allow for feedback to be given to bidders in a very natural way. Further exploring coalitional feedback as a bidder tool is a promising avenue of research, as it both reduces the complexity bidders face with package bidding, and allows them to focus more on relevant packages as well as guide their pricing decisions, considering both the deadness/winning levels as well as a number of coalitions.

One major hurdle in setting up laboratory experiments with human bidders participating in combinatorial auctions, is a technical one. The effort required to set up such an experiment is significant. While the field of experimental economics is becoming increasingly important, the effort barrier may be too large for many interested researchers. The complexity of implementing such auctions far exceeds that what is usually done in experimental economics, as it requires knowledge of several fields (statistics, programming, experimental economics, etc.). Luckily there is a fine software package (freely) available for academic use: z-Tree. However, while this package is flexible, it still requires a lot of programming to be able to run an actual combinatorial auction. There is a need for a similar toolbox to z-Tree, that can easily be adapted to the experiments combinatorial auction needs.

5.3 Practice

We contribute to practice in Chapter 4, where we describe how we apply our three phase scheduling method to create the schedules of four different conferences, MathSport 2013, MAPSP 2015 & 2017 and ORBEL 2017.

A possible consequence of our method could be that the resulting sessions are incoherent, since their composition is based solely on participant preferences, and not on the topic of the talk. However, as the profiles of the participants tend to contain talks on similar topics, the resulting sessions are still relatively coherent.

Another concern is that the current approach does not treat all participants equally. By preferring many talks, a participant may have a larger impact on the conference schedule than a participant with a small number of preferences. This can be remedied by giving an appropriate weight to the preferences of each participant, or limiting the number of preferences each participant can express to e.g. the number of time slots of the conference.

A final issue is the scalability of our approach. Although our method has been developed for medium-size conferences, the question arises to what extent it scales to much larger conferences. This is relevant both when eliciting participants' preferences, as well as computationally. The latter can be accommodated by solving the second phase using our heuristic approach, as the bottleneck appears to be the computation of the hopping coefficient for all k-blocks, rather than solving the IP formulations. In case preference elicitation is unpractical due to the high amount of talks, our approach could be applied on the level of streams or tracks. Indeed, talks in large conferences are often from the beginning (i.e., when submitting an abstract) assigned to streams, and the conference schedule is typically based on track segmentation. As not all streams cover the full length of the conference, there would be possibilities to minimize overlap between pairs of streams that many participants would like to attend. Furthermore, if overlap is unavoidable, the streams could be allocated to rooms which are close to each other, facilitating session hopping. In fact, eliciting stream preferences would provide a good idea of the required room capacities for each stream.

Appendices

A Experiment instructions

In this section we present the instructions used in the laboratory experiments, which include screenshots showing the graphical user interface subjects used during the experiments.

Introduction and goals of the study

This is an experiment in the economics of decision making. The goal of this research is to experimentally study decision behavior in strategic environments. The purpose is to apply the gained insights to better design allocation mechanisms, improving social efficiency.

The instructions are simple, and if you follow them carefully, you can, depending on your decisions, earn a considerable amount of money. This amount will be paid to you in cash later; you will be informed of the details by e-mail. It is very important that you read these instructions with care.

Procedure

The procedure of this experimental session is as follows:

1. Reading of the instructions
2. Filling in the test questions
3. Participating in a series of auctions
4. Filling in a questionnaire

The auction environment

In this experiment, we will create auction environments consisting of 9 participants; yourself along with 8 other participants. You will act as participants in a sequence of auctions. In each auction 6 items are put up for sale simultaneously, and participants may submit bids on any *combination* of items they want (also called "*package*").

There is no relation between consecutive auctions. In other words, each auction will be completely independent of the previous auction(s).

The values

Each participant will be assigned private values for all possible combinations of items. These values represent the value of the (combination of) item(s) to you. By clicking on a package, your private value for this selection will appear on the screen (see screenshot). You are not to reveal this information to any other participant. It is your own private information.

It is possible the value of a package of items is greater than the sum of the values of the items in the package separately. However, this can only occur between adjacent items.

For example, suppose the value of item 1, on its own, is 20. The value of item 2, on its own, is 40. It is possible that the value for the package {1,2} is equal to 100. Here, the value of the package (100) is larger than the sum of the values of the individual items (20+40=60).

Bids:

Each participant can bid on any combination of items. Participants are allowed to place multiple bids every round. You are free to bid whatever you think will bring you the most earnings, as long as this amount does not exceed your private valuation. There is a minimum bid increment of 1 in place. For example, you may bid a price of 26, but not of 26.3.

Once you press the 'Enter Bid'-button, the bid is finalized; it can no longer be changed or retracted.

Round structure:

Every auction consists of successive rounds in which participants may place bids. A round ends when all bidders indicate that they are done entering bids for that round, or when the time limit (in seconds) is reached. Information on the round number and the remaining time can be found at the top of the bidding screen.

Determination of winning bids:

The winning bids are selected in such a way that the total revenue (sum of the prices of the winning bids) for the round is maximized, while making sure every item is sold at most once. Note that this means one participant could possibly win more than one bid. In case of ties among the highest bids we will randomly pick a winner of the item.

Closing rule:

The auction closes after three consecutive rounds in which the total auction revenue does not increase. Only bids that are winning after the auction has closed are used to calculate auction earnings.

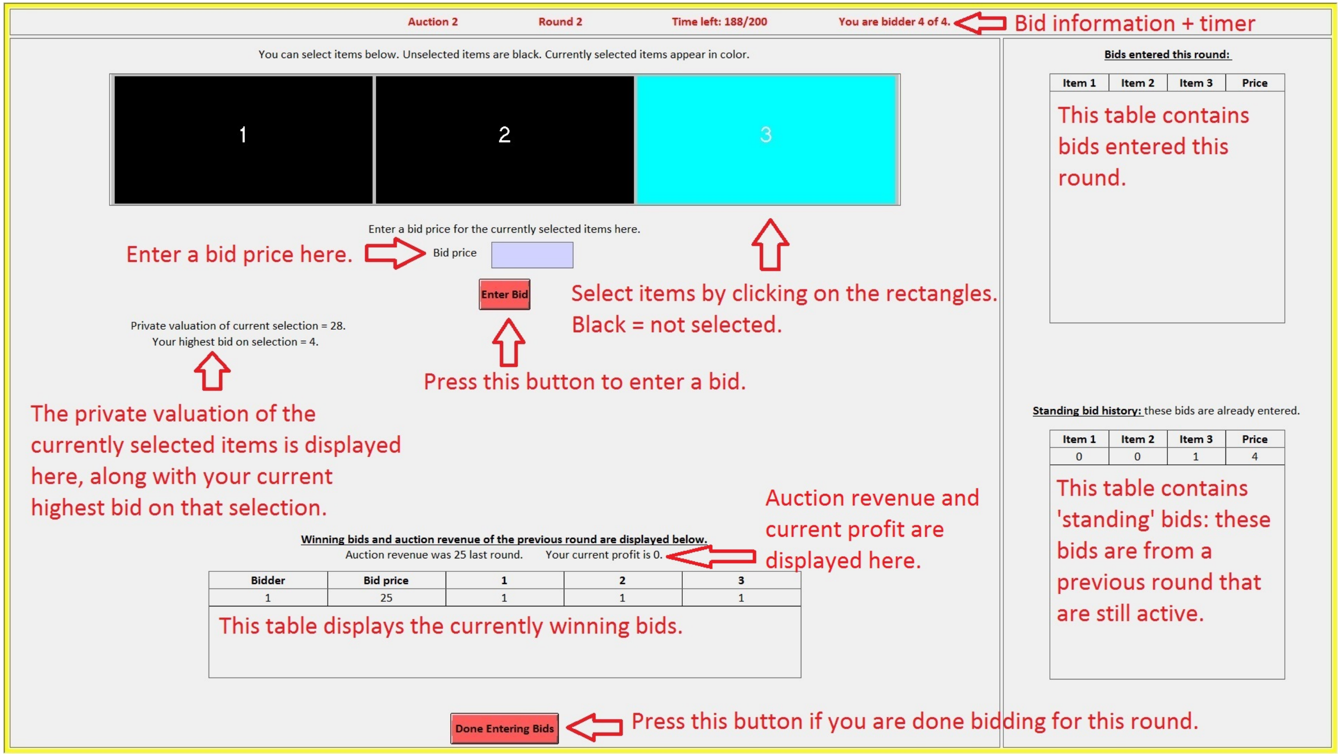
Bidder earnings:

Participant earnings depend on the results of the auctions. Winning any (combination of) item(s) at a price below the private value of those items, results in a profit. The larger this difference, the bigger the profit. After the auctions are completed, the bidder earnings will be calculated in detail by the experimenter.

The interface:

Screenshots of the bidder interface, along extra information, are given in the next 3 pages. Please examine them carefully. The first two images are from during a bidding round: here bids can entered.

The third image is from after a bidding round: it displays the results of a bidding round.



Auction 2 Round 2 Time left: 188/200 You are bidder 4 of 4.

Bid information + timer

You can select items below. Unselected items are black. Currently selected items appear in color.

Select items by clicking on the rectangles. Black = not selected.

Enter a bid price for the currently selected items here.

Enter a bid price here.

Bid price

Press this button to enter a bid.

Private valuation of current selection = 28.
Your highest bid on selection = 4.

The private valuation of the currently selected items is displayed here, along with your current highest bid on that selection.

Winning bids and auction revenue of the previous round are displayed below.

Auction revenue was 25 last round. Your current profit is 0.

Auction revenue and current profit are displayed here.

Bidder	Bid price	1	2	3
1	25	1	1	1

This table displays the currently winning bids.

Press this button if you are done bidding for this round.

Bids entered this round:

Item 1	Item 2	Item 3	Price

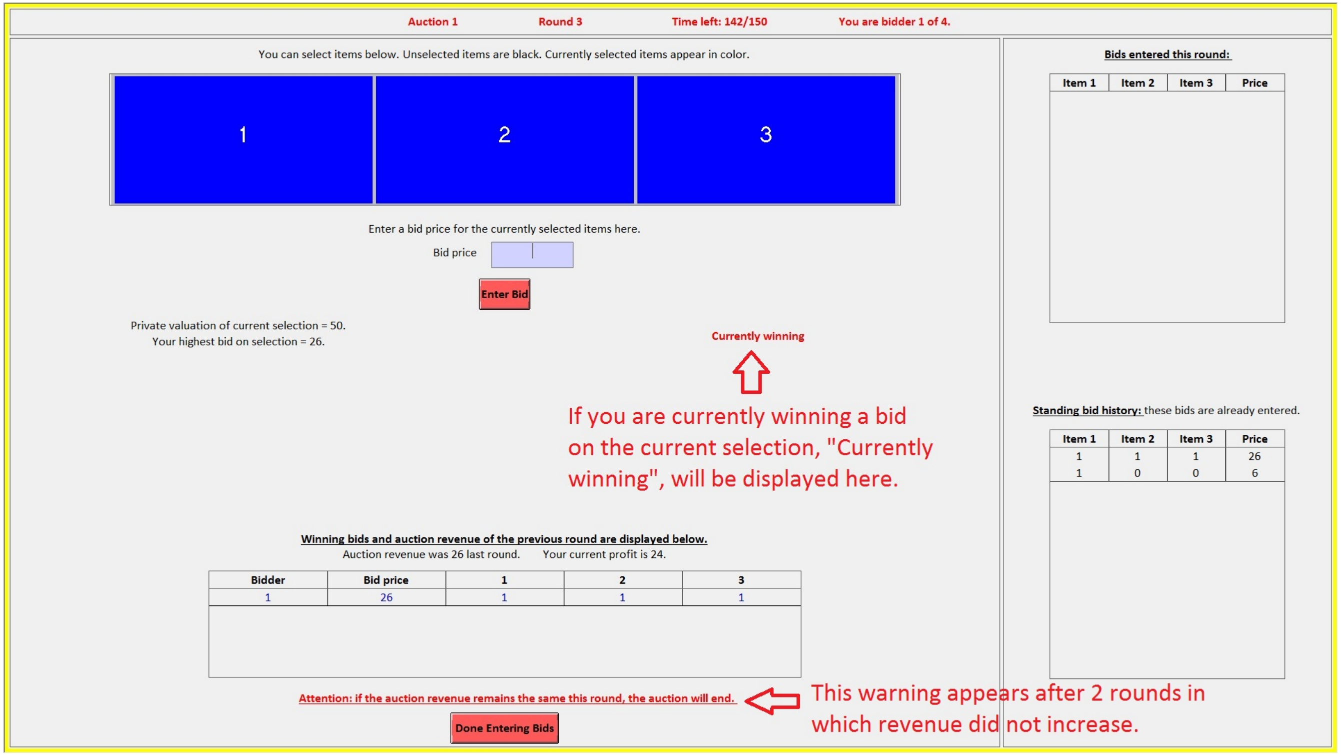
This table contains bids entered this round.

Standing bid history: these bids are already entered.

Item 1	Item 2	Item 3	Price
0	0	1	4

This table contains 'standing' bids: these bids are from a previous round that are still active.

Figure A.1: GUI: the bid interface.



Auction 1 Round 3 Time left: 142/150 You are bidder 1 of 4.

You can select items below. Unselected items are black. Currently selected items appear in color.

Enter a bid price for the currently selected items here.

Bid price

Private valuation of current selection = 50.
Your highest bid on selection = 26.

Currently winning

If you are currently winning a bid on the current selection, "Currently winning", will be displayed here.

Winning bids and auction revenue of the previous round are displayed below.

Auction revenue was 26 last round. Your current profit is 24.

Bidder	Bid price	1	2	3
1	26	1	1	1

Attention: if the auction revenue remains the same this round, the auction will end.

This warning appears after 2 rounds in which revenue did not increase.

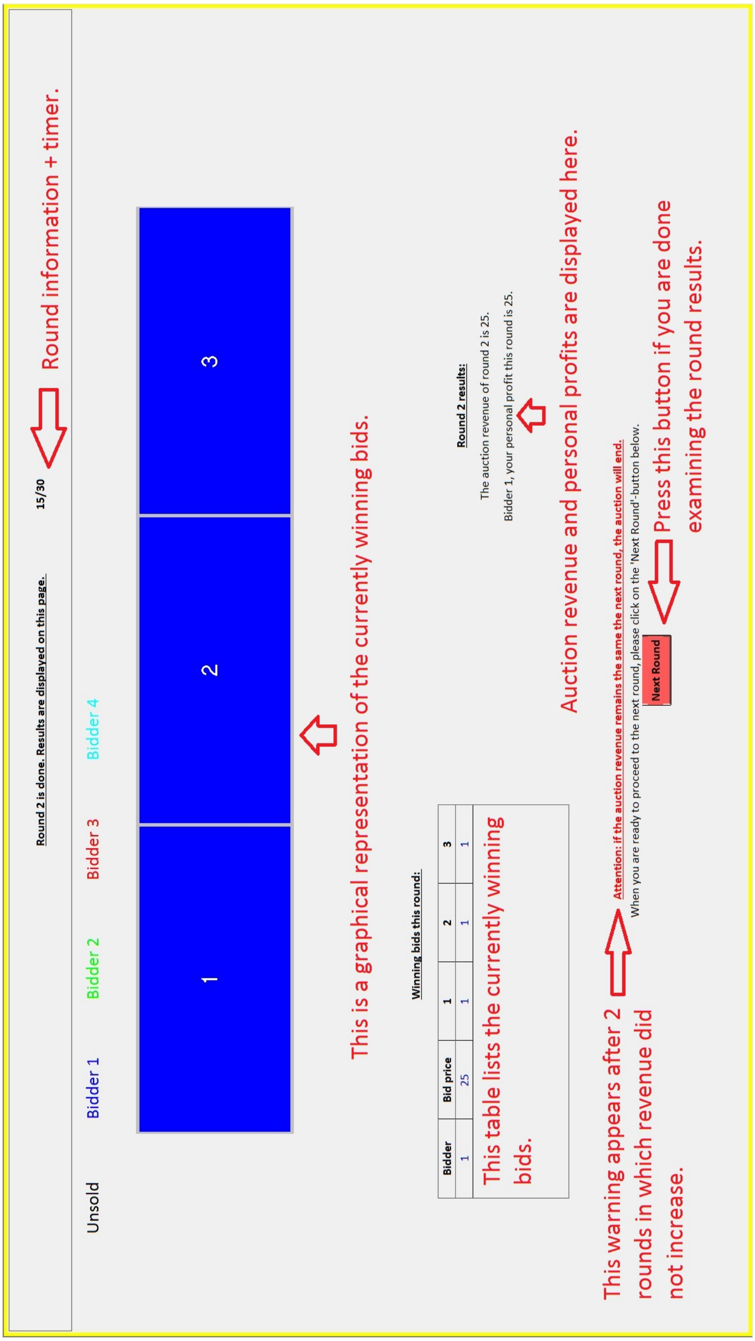
Bids entered this round:

Item 1	Item 2	Item 3	Price

Standing bid history: these bids are already entered.

Item 1	Item 2	Item 3	Price
1	1	1	26
1	0	0	6

Figure A.2: GUI: when bidders try to bid on a package they are currently (provisionally) winning.



Round 2 is done. Results are displayed on this page.

Round information + timer.

This is a graphical representation of the currently winning bids.

Winning bids this round:

Bidder	Bid price	1	2	3
1	25	1	1	1

This table lists the currently winning bids.

Round 2 results:

The auction revenue of round 2 is 25.

Bidder 1, your personal profit this round is 25.

Auction revenue and personal profits are displayed here.

Attention: if the auction revenue remains the same the next round, the auction will end.

When you are ready to proceed to the next round, please click on the 'Next Round'-button below.

This warning appears after 2 rounds in which revenue did not increase.

Press this button if you are done examining the round results.

Figure A.3: GUI: display of (provisional) winning bids after a round has finished.

Test questions:

After reading the instructions, please fill in the questions below. The experimenter will go over the answers to the test questions before starting the auctions.

- Suppose your valuation for some package of items is equal to 8. What is the maximum price you can bid on this set of items?

  . . .

- Suppose the revenue of round 4 was 30. Will there be a next round if the revenue of round 5 will be 30?

  . . .

- Suppose the revenue of round 4 was 30. The revenue of round 5 was 30. Will there be a next round if the revenue of round 6 will be 30?

  . . .

- Suppose you enter a bid on the package {1,2}. Are you allowed to also enter a bid on the package {2,3} in the same round?

  . . .

- Suppose you did not enter a bid on the package {1,2} in round 1. Are you allowed to enter a bid on the package {1,2} in any following round?

  . . .

- Suppose you enter a bid on the package {2,3}. Are you allowed to enter another bid on the package {2,3} in the same round?

  . . .

- After a round, is it possible that you win multiple bids?

  . . .

B Feedback explanations

This section contains the explanation used in the laboratory experiments for feedback level 4 (FB4).

Winning levels

After round 1, every unique combination of selected items will have its own winning level. The winning level represents the price required for a bid to win, given that all other bids remain the same. Note: even if you are provisionally winning a bid, you will still get winning level feedback.

Example 1: you select items 1 and 2 in round 5. The displayed winning level is 50. If you bid 50 on items 1 and 2, and nobody else makes a bid this round, you are sure to win items 1 and 2 for 50.

Example 2: you select items 2 and 3 in round 4. The displayed winning level is 45. If you bid 40 on items 2 and 3, and nobody else makes a bid this round, your bid of 40 on items 2 and 3 will not become winning.

Example 3: you select items 1, 2, and 3 in round 6. The displayed winning level is 100. Other bidders entered bids in round 6. Even if you bid more than 100 on items 1, 2, and 3, there is no guarantee your bid will become winning.

Example 4: in round 3, you have a provisional winning bid on item 1 with a price of 75. In round 4, if you select item 1, the displayed winning level will be 75.

Deadness levels

After round 1, every unique combination of selected items will have its own deadness level. The deadness level represents the minimum price required for a bid to ever be able to become winning, given that all other bids remain the same.

Example 1: you select items 1 and 2 in round 5. The displayed deadness level is 50. If you bid 50 on items 1 and 2, and nobody else makes a bid this round, your bid could still become winning in a future round.

Example 2: you select items 2 and 3 in round 4. The displayed deadness level is 45. If you bid 40 on items 2 and 3, your bid of 40 on items 2 and 3 will not become winning.

Example 3: you select items 1, 2, and 3 in round 6. The displayed deadness level is 100. Even if you bid more than 100 on items 1, 2, and 3, there is no guarantee your bid will ever become winning.

Coalitional feedback with suggestion

Coalitional feedback can appear if there are multiple bids that can become winning together, thus beating the currently winning bids, given that all bids outside of the coalition remain the same. Coalitional feedback is, when available, given in the following format:

"If x bids, including this one, are collectively raised by y, these x bids become winning. We suggest you bid z."

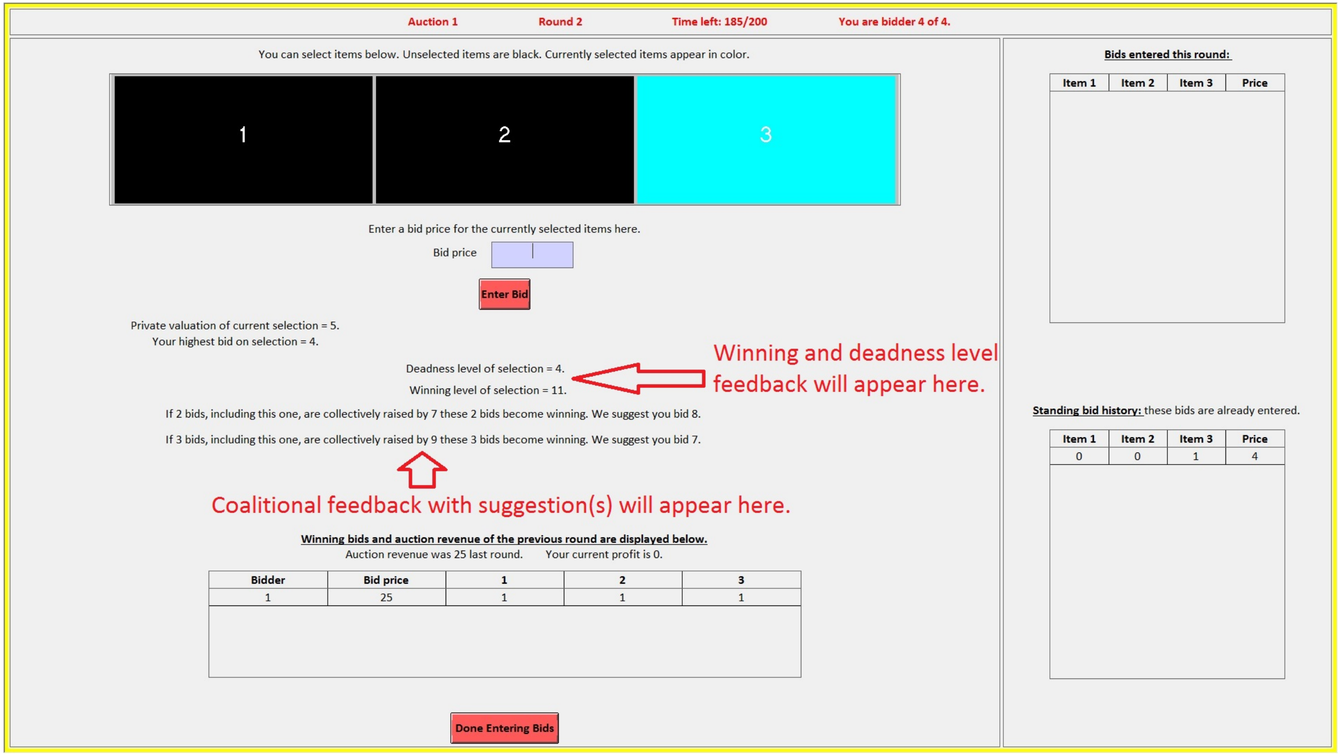
The word collectively is important here: all x bids have to increase their current bids by a total of y to beat the currently winning allocation. If every bid is increased to z, as suggested, those bids become winning together.

All bids in a coalition receive a similar message.

Note that it is possible to receive multiple such messages for a single bid.

The interface

A screenshots of the bidder interface and where the feedback is located is given on the following page.



You can select items below. Unselected items are black. Currently selected items appear in color.

Enter a bid price for the currently selected items here.

Bid price

Private valuation of current selection = 5.
Your highest bid on selection = 4.

Deadness level of selection = 4.
Winning level of selection = 11.

If 2 bids, including this one, are collectively raised by 7 these 2 bids become winning. We suggest you bid 8.

If 3 bids, including this one, are collectively raised by 9 these 3 bids become winning. We suggest you bid 7.

Winning and deadness level feedback will appear here.

Coalitional feedback with suggestion(s) will appear here.

Winning bids and auction revenue of the previous round are displayed below.
Auction revenue was 25 last round. Your current profit is 0.

Bidder	Bid price	1	2	3
1	25	1	1	1

Bids entered this round:

Item 1	Item 2	Item 3	Price

Standing bid history: these bids are already entered.

Item 1	Item 2	Item 3	Price
0	0	1	4

Figure B.1: GUI: FB4 message for a bid.

C Design of experiment

The experimental design is presented in Table C.1. A total of 192 auctions were held. An experimental session consists of 4 groups, one for every level of the factor Structure. Every group in an experimental session is called an experimental unit. An experimental unit consists of a series of 4 consecutive auctions with the same participants, and there is 1 auction per unique feedback level. Every session requires 27 subjects (one experimental unit consisting of 4 subjects, two experimental units consisting of 7 subjects, and one experimental unit consisting of 9 subjects), hence the total number of required participants for 12 sessions is 324.

The design is between-subject for the factor Structure, and within-subject for the factors threshold and feedback. In addition, all 24 permutations of the 4 feedback levels occur exactly twice, all threshold levels occur at least once per experimental unit and any two consecutive threshold levels within an experimental unit are distinct. Unfortunately, due to a programming error the balance in STR4 was lost. Specifically, in the STR4 experimental units, every time a CT level of 3 occurs, in reality a private value set with a CT value of 2 was used. In other words, we test more 'difficult' threshold cases, and less 'easy' threshold cases for STR4. These occurrences are indicated in the table using boldface.

Session	STR1				STR2				STR3				STR4			
1	1;1	2;2	1;3	3;4	2;3	3;4	1;1	3;2	1;4	3;3	1;2	2;1	1;3	2;4	**2**;2	2;1
2	1;2	2;3	1;4	3;1	2;1	1;3	3;2	2;4	1;2	2;4	3;3	1;1	2;2	1;4	**2**;1	1;3
3	1;3	3;4	2;1	3;2	2;4	3;2	2;3	1;1	3;4	2;3	3;1	1;2	1;2	2;1	**2**;3	1;4
4	1;4	3;2	1;1	2;3	3;1	2;4	1;3	2;2	2;1	1;4	2;3	3;2	1;1	**2**;3	1;2	2;4
5	2;1	1;2	3;3	2;4	2;2	1;4	3;1	1;3	2;2	1;3	2;1	3;4	**2**;1	2;3	**2**;4	1;2
6	2;2	1;1	2;4	3;3	3;2	2;1	3;3	1;4	1;3	3;2	1;1	2;4	1;4	**2**;1	1;3	2;2
7	2;3	3;1	2;2	1;4	3;3	1;2	2;1	3;4	1;1	3;4	2;2	3;3	2;1	**2**;2	1;4	2;3
8	2;4	3;3	1;2	2;1	3;4	1;1	2;2	3;3	3;2	2;1	1;4	2;3	2;3	**2**;4	2;2	1;1
9	3;1	2;4	3;2	1;3	1;2	2;3	1;4	3;1	2;3	1;1	3;2	1;4	2;4	**2**;2	1;1	**2**;3
10	3;2	1;4	2;3	1;1	1;4	3;1	1;2	2;3	3;1	2;2	3;4	1;3	**2**;3	1;1	2;4	**2**;2
11	3;3	2;1	3;4	1;2	1;3	2;2	3;4	2;1	2;4	3;1	1;3	2;2	2;4	1;2	2;3	**2**;1
12	3;4	1;3	3;1	2;2	1;1	3;3	2;4	1;2	3;3	1;2	2;4	3;1	**2**;2	1;3	2;1	**2**;4

Table C.1: Experimental sessions: first number in an experimental unit represents the CT level, second number represents the FB level.

Bibliography

Abrache, J., Crainic, T. G., Gendreau, M., and Rekik, M. (2007). Combinatorial auctions. *Annals of Operations Research*, 153(1):131–164.

Adomavicius, G., Curley, S. P., Gupta, A., and Sanyal, P. (2012). Effect of Information Feedback on Bidder Behavior in Continuous Combinatorial Auctions. *Management Science*, 58(4):811–830.

Adomavicius, G. and Gupta, A. (2005). Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions. *Information Systems Research*, 16(2):169–185.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Ausubel, L. and Milgrom, P. (2002). Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1):1019–1+.

Ausubel, L. M. and Milgrom, P. (2006). *The lovely but lonely Vickrey auction*, chapter 1, pages 17–40. Combinatorial Auctions. MIT Press, Cambridge, Massachussetts.

Babaioff, M. and Blumrosen, L. (2008). Computationally-feasible truthful auctions for convex bundles. *Games and Economic Behavior*, 63(2):588–620.

Banks, J., Olson, M., Porter, D., Rassenti, S., and Smith, V. (2003). Theory, experiment and the federal communications commission spectrum auctions. *Journal of Economic Behavior & Organization*, 51(3):303–350.

Banks, J. S., Ledyard, J. O., and Porter, D. P. (1989). Allocating Uncertain and Unresponsive Resources: An Experimental Approach. *The RAND Journal of Economics*, 20(1):1–25.

Bar-Yehuda, R., Halldórsson, M. M., Naor, J., Shachnai, H., and Shapira, I. (2006). Scheduling Split Intervals. *SIAM Journal on Computing*, 36(1):1–15.

Berman, F., Johnson, D., Leighton, T., Shor, P. W., and Snyder, L. (1990). Generalized Planar Matching. *Journal of Algorithms*, 11(2):153–184.

Bichler, M., Hao, Z., and Adomavicius, G. (2017). Coalition-based pricing in ascending combinatorial auctions. Submitted to Informations Systems Research.

Bichler, M., Shabalin, P., and Wolf, J. (2013). Do core-selecting Combinatorial Clock Auctions always lead to high efficiency? An experimental analysis of spectrum auction designs. *Experimental Economics*, 16(4):511–545.

Brunner, C., Goeree, J. K., Holt, C. A., and Ledyard, J. O. (2010). An Experimental Test of Flexible Combinatorial Spectrum Auction Formats. *American Economic Journal: Microeconomics*, 2(1):39–57.

Buchfuhrer, D., Schapira, M., and Singer, Y. (2010). Computation and incentives in combinatorial public projects. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 33–42. ACM.

Bykowsky, M. M., Cull, R. J., and Ledyard, J. O. (2000). Mutually Destructive Bidding: The FCC Auction Design Problem. *Journal of Regulatory Economics*, 17(3):205–228.

Cantillon, E. and Pesendorfer, M. (2006). *Auctioning bus Routes: The London Experience*, chapter 22, pages 573–592. Combinatorial Auctions. MIT Press, Cambridge, Massachussetts.

Chalermsook, P. and Chuzhoy, J. (2009). Maximum Independent Set of Rectangles. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 892–901.

Chan, T. M. and Har-Peled, S. (2012). Approximation Algorithms for Maximum Independent Set of Pseudo-Disks. *Discrete Computational Geometry*, 48:373–392.

Chernomaz, K. and Levin, D. (2012). Efficiency and synergy in a multi-unit auction with and without package bidding: An experimental study. *Games and Economic Behavior*, 76(2):611–635.

Christodoulou, G., Elbassioni, K., and Fouz, M. (2010). Truthful Mechanisms for Exhibitions. In *Internet and Network Economics*, volume 6484, pages 170–181. Springer Berlin Heidelberg.

Cramton, P. (2007). How Best to Auction Oil Rights. In Humphreys, M., Sachs, J., and Stiglitz, J. E., editors, *Escaping the resource curse*, chapter 5, pages 114–151. Cambridge Univ Press.

Cramton, P. and Schwartz, J. (2000). Collusive Bidding: Lessons from the FCC Spectrum Auctions. *Journal of Regulatory Economics*, 17(3):229–252.

Cramton, P., Shoham, Y., and Steinberg, R., editors (2006). *Combinatorial auctions.* MIT Press.

Day, R. W. and Raghavan, S. (2008). A combinatorial procurement auction featuring bundle price revelation without free-riding. *Decision Support Systems*, 44(3):621–640.

de Vries, S. and Vohra, R. V. (2003). Combinatorial Auctions: A Survey. *INFORMS Journal on Computing*, 15(3):284–309.

Edis, E. and Sancar Edis, R. (2013). An integer programming model for the conference timetabling problem. *Celal Bayal University Journal of Science*, 9(2):55–62.

Eglese, R. and Rand, G. (1987). Conference seminar timetabling. *Journal of the Operational Research Society*, 38(7):591–598.

Epstein, R., Henríquez, L., Catalán, J., Weintraub, G. Y., and Martínez, C. (2002). A Combinational Auction Improves School Meals in Chile. *Interfaces*, 32(6):1–14.

Erlebach, T., Jansen, K., and Seidel, E. (2001). Polynomial-time Approximation Schemes for Geometric Graphs. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 671–679.

Ernst, A. T., Mills, R., and Welgama, P. (2003). Scheduling appointments at trade events for the australian tourist commission. *Interfaces*, 33(3):12–23.

Ernst, A. T., Singh, G., and Weiskircher, R. (2008). Scheduling meetings at trade events with complex preferences. In *Proceedings of 18th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 76–82.

Fischbacher, U. (2007). z-tree: Zurich toolbox for ready-made economic experiments. *Experimental Economics*, 10(2):171–178.

Fox, J. T. and Bajari, P. (2013). Measuring the Efficiency of an FCC Spectrum Auction. *American Economic Journal: Microeconomics*, 5(1):100–146.

Goeree, J. K. and Holt, C. A. (2010). Hierarchical package bidding: A paper & pencil combinatorial auction. *Games and Economic Behavior*, 70(1):146–169. Special Issue In Honor of Ehud Kalai.

Goossens, D. R., Onderstal, S., Pijnacker, J., and Spieksma, F. C. R. (2014). Solids: A Combinatorial Auction for Real Estate. *Interfaces*, 44(4):351–363.

Grant, P. (2014). The dilemma of attending (or not) scientific conferences. *Canadian Journal of Physiology and Pharmacology*, 92(1):v–v.

Gremillet, D. (2008). Paradox of flying to meetings to protect the environment. *Nature*, 455(7217):1175–1175.

Gulati, M. and Sengupta, A. (2004). TRACS: Tractable Conference Scheduling. In *Proceedings of the Decision Sciences Institute Annual Meeting (DSI 2004)*, pages 3161–3166.

Günlük, O., Ladányi, L., and de Vries, S. (2005). A Branch-and-Price Algorithm and New Test Problems for Spectrum Auctions. *Management Science*, 51(3):391–406.

Hochbaum, D. S. and Maass, W. (1985). Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI. *The Journal of the ACM*, 32(1):130–136.

Ibrahim, H., Ramli, R., and Hassan, M. (2008). Combinatorial design for a conference: constructing a balanced three-parallel session schedule. *Journal of Discrete Mathematical Sciences & Cryptography*, 11(3):305–317.

Ignatius, J., Hosseini-Motlagh, S.-M., Goh, M., Sepehri, M. M., Mustafa, A., and Rahman, A. (2014). Multiobjective combinatorial auctions in transportation procurement. *Mathematical Problems in Engineering*, 2014:1–9.

Ioannidis, J. P. (2012). Are medical conferences useful? and for whom? *Journal of the American Medical Association*, 307(12):1257–1258.

Jackson, C. (1976). *Technology for Spectrum Markets*. PhD thesis, Department of Electrical Engineering, School of Engineering, MIT.

Kagel, J. H., Lien, Y., and Milgrom, P. (2010). Ascending Prices and Package Bidding: A Theoretical and Experimental Analysis. *American Economic Journal: Microeconomics*, 2(3):160–85.

Kagel, J. H., Lien, Y., and Milgrom, P. (2014). Ascending prices and package bidding: Further experimental analysis. *Games and Economic Behavior*, 85(0):210–231.

Kazumori, E. (2010). Core-Selecting Auctions: An Experimental Study. CARF F-Series CARF-F-226, Center for Advanced Research in Finance, Faculty of Economics, The University of Tokyo.

Kwasnica, A. M., Ledyard, J. O., Porter, D., and DeMartini, C. (2005). A New and Improved Design for Multiobject Iterative Auctions. *Management Science*, 51(3):419–434.

Le Page, Y. (1996). Optimized schedule for large crystallography meetings. *Journal of Applied Crystallography*, 29(3):291–295.

Leung, J. Y. T., Tam, T. W., Wong, C., Young, G. H., and Chin, F. Y. (1990). Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3):271–275.

Lovász, L. and Plummer, M. D. (1986). *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. AMS Chelsea Publishing.

Lucier, B., Singer, Y., Syrgkanis, V., and Tardos, É. (2013). Equilibrium in combinatorial public projects. In Chen, Y. and Immorlica, N., editors, *Web and Internet Economics: 9th International Conference, WINE 2013*, Lecture Notes in Computer Science, pages 347–360.

Markakis, E. and Telelis, O. (2017). Item bidding for combinatorial public projects. *Theoretical Computer Science*, 678:40–55.

Martin, A. (2005). A personalised conference programme advisor. *OR Insight*, 18(2):22–30.

McMillan, J. (1994). Selling Spectrum Rights. *The Journal of Economic Perspectives*, 8(3):145–162.

Mu'alem, A. and Nisan, N. (2008). Truthful approximation mechanisms for restricted combinatorial auctions. *Games and Economic Behavior*, 64(2):612–631. Special Issue in Honor of Michael B. Maschler.

Nathans, J. and Sterling, P. (2016). Point of view: How scientists can reduce their carbon footprint. *eLIFE*.

Nicholls, M. (2007). A small-to-medium-sized conference scheduling heuristic incorporating presenter and limited attendee preferences. *Journal of the Operational Research Society*, 58(3):301–308.

Nisan, N. (2000). Bidding and Allocation in Combinatorial Auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, EC '00, pages 1–12.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. (2007). *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge.

Papadimitriou, C., Schapira, M., and Singer, Y. (2008). On the hardness of being truthful. In *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '08, pages 250–259.

Parkes, D. C. and Ungar, L. H. (2000). Iterative Combinatorial Auctions: Theory and Practice. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 74–81.

Petrakis, I., Ziegler, G., and Bichler, M. (2013). Ascending Combinatorial Auctions with Allocation Constraints: On Game Theoretical and Computational Properties of Generic Pricing Rules. *Information Systems Research*, 24(3):768–786.

Plott, C. R. and Salmon, T. C. (2004). The simultaneous, ascending auction: dynamics of price adjustment in experiments and in the UK3G spectrum auction. *Journal of Economic Behavior & Organization*, 53(3):353–383.

Porter, D., Rassenti, S., Roopnarine, A., and Smith, V. (2003). Combinatorial auction design. *Proceedings of the National Academy of Sciences*, 100(19):11153–11157.

Potthoff, R. and Brams, S. (2007). Scheduling of panels by integer programming: Results for the 2005 and 2006 New Orleans meetings. *Public Choice*, 131(2):465–468.

Potthoff, R. and Munger, M. (2003). Use of integer programming to optimize the scheduling of panels at annual meetings of the Public Choice Society. *Public Choice*, 117(1):163–175.

Quan, D. (1994). Real Estate Auctions: A Survey of Theory and Practice. *Journal of Real Estate Finance and Economics*, 9(1):23–49.

Quesnelle, J. and Steffy, D. (2015). Scheduling a conference to minimize attendee preference conflicts. In Hanzálek, Z., Kendall, G., McCollum, B., and Šucha, P., editors, *Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)*, pages 379–392.

Rassenti, S. J., Smith, V. L., and Bulfin, R. L. (1982). A combinatorial mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417.

Rendl, F. and Woeginger, G. (1993). Reconstructing sets of orthogonal line segments in the plane. *Discrete Mathematics*, 119(1-3):167–174.

Rothkopf, M. H., Pekeč, A., and Harstad, R. M. (1998). Computationally Manageable Combinational Auctions. *Management Science*, 44(8):1131–1147.

Sampson, S. (2004). Practical Implications of Preference-Based Conference Scheduling. *Production and Operations Management*, 13(3):205–215.

Sampson, S. and Weiss, E. (1995). Increasing service levels in conference and educational scheduling: a heuristic approach. *Management Science*, 41(11):1816–1825.

Sampson, S. and Weiss, E. (1996). Designing Conferences to Improve Resource Utilization and Participant Satisfaction. *Journal of the Operational Research Society*, 47(2):297–314.

Sandholm, T. (2002). Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54.

Sandholm, T., Suri, S., Gilpin, A., and Levine, D. (2005). CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. *Management Science*, 51(3):374–390.

Schapira, M. and Singer, Y. (2008). Inapproximability of combinatorial public projects. In Papadimitriou, C. and Zhang, S., editors, *Internet and Network Economics: 4th International Workshop, WINE 2008*, Lecture Notes in Computer Science, pages 351–361.

Scheffel, T., Pikovsky, A., Bichler, M., and Guler, K. (2011). An Experimental Comparison of Linear and Nonlinear Price Combinatorial Auctions. *Information Systems Research*, 22(2):346–368.

Scheffel, T., Ziegler, G., and Bichler, M. (2012). On the impact of package selection in combinatorial auctions: an experimental study in the context of spectrum auction design. *Experimental Economics*, 15(4):667–692.

Seifert, S. and Ehrhart, K.-M. (2005). Design of the 3G Spectrum Auctions in the UK and Germany: An Experimental Investigation. *German Economic Review*, 6(2):229–248.

Spieksma, F. C. R. (1999). On the approximability of an interval scheduling problem. *Journal of Scheduling*, 2:215–227.

Tanaka, M. and Mori, Y. (2002). A Hybrid Grouping Genetic Algorithm for Timetabling of Conference Programs. In De Causmaecker, P. and Burke, E., editors, *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002)*, pages 421–440.

Tanaka, M., Mori, Y., and Bargiela, A. (2002). Granulation of Keywords into Sessions for Timetabling Conferences. In *Proceedings of Soft Computing and Intelligent Systems (SCIS 2002)*, pages 1–5.

Thompson, G. (2002). Improving Conferences through Session Scheduling. *Cornell Hotel and Restaurant Administration Quarterly*, 43(3):71–76.

van Bevern, R., Mnich, M., Niedermeier, R., and Weller, M. (2015). Interval scheduling and colorful independent sets. *Journal of Scheduling*, 18:449–469.

Van Hoesel, S. and Müller, R. (2001). Optimization in electronic markets: examples in combinatorial auctions. *Netnomics*, 3(1):23–33.

van Rooij, J. M. M., van Kooten Niekerk, M. E., and Bodlaender, H. L. (2012). Partition Into Triangles on Bounded Degree Graphs. *Theory of Computing Systems*, 52(4):687–718.

Vangerven, B., Ficker, A. M., Goossens, D. R., Passchyn, W., Spieksma, F. C., and Woeginger, G. J. (2017a). Conference scheduling - a personalized approach. *Omega*.

Vangerven, B., Goossens, D. R., and Spieksma, F. C. (2017b). Winner determination in geometrical combinatorial auctions. *European Journal of Operational Research*, 258(1):254–263.

Vialette, S. (2004). On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science*, 312(2):223–249.

Xia, M., Stallaert, J., and Whinston, A. B. (2005). Solving the combinatorial double auction problem. *European Journal of Operational Research*, 164(1):239–251.

Yingping, H., Zhang, X., and Alexander, P. S. (2012). A heuristic algorithm for optimizing business matchmaking scheduling. *International Journal of Operations Research and Information Systems*, 3.4:59–73.

Zulkipli, F., Ibrahim, H., and Benjamin, A. (2013). Optimization Capacity Planning Problem on Conference Scheduling. In *Proceedings of the Business Engineering and Industrial Applications Colloquium (BEIAC 2013)*, pages 911–915.