FACULTEIT ECONOMISCHE EN
TOEGEPASTE ECONOMISCHE
WETENSCHAPPEN

KATHOLIEKE
UNIVERSITEIT
LEUVEN

EXACT METHODS FOR COMBINATORIAL AUCTIONS

Proefschrift voorgedragen
tot het behalen van de graad
van Doctor in de Toegepaste
Economische Wetenschappen

door

Dries Goossens

Nummer 245 2006

Committee

Prof. dr. Frits C.R. Spieksma (Advisor)	*Katholieke Universiteit Leuven*
Prof. dr. Dirk Cattrysse	*Katholieke Universiteit Leuven*
Prof. dr. Sven de Vries	*Technische Universität München*
Prof. dr. Willy Gochet	*Katholieke Universiteit Leuven*
Prof. dr. Willy Herroelen	*Katholieke Universiteit Leuven*
Prof. dr. Rudolf Müller	*Universiteit Maastricht*

Daar de proefschriften in de reeks van de Faculteit Economische en Toegepaste Economische Wetenschappen het persoonlijk werk zijn van hun auteurs, zijn alleen deze laatsten daarvoor verantwoordelijk.

Dankwoord

Het schrijven van een doctoraat is geen sinecure. Ik kan mij inbeelden dat iemand die het alleen zou moeten doen, vaak met de handen in het haar zou zitten en het licht aan het eind van de tunnel niet meer zou zien. Gelukkig stond ik er niet alleen voor, en kan ik in dit dankwoord de mensen die mij geholpen hebben bedanken.

Eerst en vooral wil ik mijn promotor Frits Spieksma van harte bedanken. Hij gaf me de kans om aan dit doctoraat te beginnen en het is ook voor een niet gering stuk zijn verdienste dat het tot een goed einde gekomen is. Bedankt Frits, voor de ontelbare interessante gesprekken, voor het geven van richting, voor het vertrouwen, en voor het inspireren. Bedankt ook om mij in dit alles toch veel vrijheid te gunnen, bijvoorbeeld om mij op voetbalkalenders en megabike toe te leggen. Ik ben ook dankbaar voor de vele blijken van begrip toen mijn hoofd niet echt naar onderzoek stond. Ik zal dit niet gauw vergeten.

Verder zou ik ook graag alle leden van mijn doctoraatscommissie willen bedanken. Er bestaat geen twijfel over dat hun waardevolle opmerkingen dit proefschrift sterk ten goede gekomen zijn. Willy Gochet was er al bij toen ik kwam solliciteren om er over te waken dat ik hier toch niet kwam "om uit te bollen". Ik hoop dat ik hem intussen van het tegendeel heb kunnen overtuigen. Laatst kreeg ik van Willy Herroelen een versie van mijn tekst terug, aangevuld met zijn commentaren. Het was een duidelijk bewijs dat er geen drie woorden van deze tekst aan zijn nauwgezet nazicht zijn ontsnapt. Ook Dirk Cattrysse wil ik danken voor het zetelen in mijn commissie en het grondig nalezen van dit werk.

Ich möchte gern meine Anerkennung äußern beim Rudolf Müller. Ich bin beehrt ihn als Mitglied meiner Jury zu haben und bin dankbar für seine Unterstützung. Ich erinnere mich auch an seine herzliche Willkommen wenn ich nach Maastricht ging um meine Arbeit zu besprechen. Ich kam zurück nach Leuven mit vielen Ideen und mehr Einsicht in Auktion Theorie. Weiterhin möchte ich mich erkenntlich zeigen beim Sven De Vries für seine enthusiastische Rückantwort auf meine Bitte Mitglied der Doktoratsjury zu werden, was in Anbetracht der Strecke die er zurücklegen müsste nicht ganz selbstverständlich war. Der Menge von Kommentaren und Emphelungen die Stirn bieten, bedeutete lange Arbeitstage für mich, aber war bestimmt ganz wertvoll. Er hat mich auch einige Auktion Spiele gelernt die so gut wie immer rentabel sind für den Auktionator.

I also wish to express my gratitude towards Bob Day. The second part of this thesis is inspired by his dissertation. It was a great pleasure for me to be able to discuss the matrix bid auction with him on conferences in San Francisco and Reykjavik. I am also grateful for his problem generator, which was used to generate the instances used in the last chapters.

Zonder mijn collegas bij ORSTAT en Productie en Logistiek zouden de voorbije vier jaar ongetwijfeld heel wat minder aangenaam geweest zijn. Ik zal altijd met plezier terugdenken aan de pauzes, de congressen die we samen meegemaakt hebben, de uitstapjes, de departementskwissen, de voetbalwedstrijdjes, ... In het bijzonder wil ik toch mijn grote voorbeelden Jan, Roel, Peter en Linda vermelden. Ik heb het misschien nooit echt laten blijken, maar ik heb altijd naar jullie opgekeken. Tenslotte wil ik ook mijn bureaugenoten Kristel en Jeroen bedanken. Kristel was vaak initiatiefneemster voor allerlei activiteiten buiten het werk, wat zeer zeker heeft bijgedragen tot de goede sfeer in onze onderzoeksgroep. Met Jeroen baat ik tegenwoordig het meest beposterde bureau van het Hogenheuvelcollege uit. Wij hebben elkaar vaak bezig gehouden met boeiende conversaties over voetbal, kolomgeneratie, Zelda, onderwijs, geocaching, de optimale gigabike ploeg, socialisten, buitenaardsen, en andere bizarre materies. Toen die gesprekken wat lang uitliepen, was het tegelijk een motivatie om eens serieus verder te werken. Wat mij betreft een ideale werkomgeving, waarvoor dank!

Een bijzonder dankjewel voor mijn ouders, voor de vaardigheden en de waarden die ze me tijdens hun opvoedingsproject bijbrachten, en de thuis waar ik elk weekend kon naar terugkeren. Zonder jullie stond ik niet waar ik nu sta.

Helemaal op het eind van dit dankwoord, maar zeker niet in het minst wil ik jou bedanken, Elke. Veel meer dan je denkt ben je een grote steun geweest voor mij, in zoveel meer dingen dan dit doctoraat. Je bent mijn grootste ontdekking.

Dries Goossens
December 2006

Abstract

This thesis focusses on combinatorial auctions. These are auctions where multiple items are for sale simultaneously to a set of buyers. Furthermore, in a combinatorial auction, a buyer is allowed to place bids on subsets of the available items. When multiple items are considered, it is often the case that the value of a set of items is higher or lower than the sum of the values of the individual items. These so-called complementarity or substitution-effects, respectively, may be bidder specific and of considerable importance. A combinatorial auction offers the bidders a way to express these effects in their bids, which may also benefit the auctioneer. This thesis mainly deals with the so-called *winner determination problem*, which consists of deciding which bids to accept and which to turn down, in order to maximize the total winning bid value. Each winning bidder thus pays his bid for the set of items he wins. In general, however, this problem is NP-hard and difficult to approximate. Numerous attempts to cope with this computational complexity can be found in literature. One approach is to impose limitations on what a bidder can bid for a set. This thesis can be divided into two parts, each dealing with a special kind of combinatorial auction, in which bidder's preferences are restricted in some way.

The first part of this thesis deals with the so-called *total quantity discount auction.* In this auction, the bidders are sellers and a bid corresponds to an offer to sell a set of items at some price. On behalf of a buyer, the auctioneer's task is to accept bids such that given amounts of different items can be

purchased at a minimal total cost. Furthermore, with his bids, each bidder grants a discount that depends on the total amount of units the buyer purchases from that bidder. More precisely, with each bid, a volume interval is specified in which the total amount of purchased items should lie if this bid is to be accepted. Furthermore, each bid has a unit price for each item. Obviously, per bidder, a bid with a higher volume interval should have lower prices, in order to result in a discount.

One theoretical result is that the winner determination problem of this auction is NP-hard. This result remains valid for some variants of this problem. Even more, our results show that the existence of a constant-factor approximation algorithm would imply $P = NP$. Another theoretical result is that the LP-relaxation of a formulation for this problem can be solved by solving a min-cost flow problem. We use this property to develop a branch-and-bound algorithm that solves a min-cost flow problem in every node of its search tree. This algorithm can also be used in four variants of the total quantity discount auction. In a first variant, the market share that one or more suppliers can obtain is constrained. Another variant allows the buyer to procure more units than strictly needed, in order to reach a lower total cost. In a third variant, the number of winning suppliers is limited, both in general and per item. Finally, we investigate a multi-period variant, where the buyer not only needs to decide what items to buy from what supplier, but also when to do this, while considering the inventory costs.

Our empirical results are based on computational experiments on instances involving up to 50 suppliers and 100 items. We compare the performance of the min-cost flow based algorithm with two other algorithms: an LP-based branch-and-bound algorithm and a branch-and-cut algorithm. It turns out that even the large instances of the basic problem are solved to optimality within a limited amount of time. We find that what algorithm performs best depends on the variant and the size of the problem.

To conclude the first part, we study so-called *exclusionary side constraints.* In the total quantity discount auction, this type of constraint enforces that only one bid per supplier can be the winning bid. We try to gain insight in how these constraints add to the complexity by studying them in the context of the transportation problem. In this context, an exclusionary side constraint on a pair of supply nodes imposes that at most one of these nodes can be used to supply a demand node. Three special cases of the transportation problem with exclusionary side constraints are considered, of which one particularly resembles the winner determination problem of the total quantity discount auction. We settle the computational complexity for all cases considered.

In the second part of this thesis, we study the so-called *matrix bid auction.* This auction limits a bidder in expressing his preferences, in the sense that the extra value an item adds to a bid on a set is determined only by the number of items in this set outranking this item, according to an ordering of the items expressed by the bidder. We show how it can be verified whether a matrix bid satisfies economic properties as free disposal, complement freeness, decreasing marginal valuations, and the gross substitutes property. Moreover, we provide a procedure to check whether a given collection of bids can be represented by a matrix bid. If this is not the case, we present a method to approximate these bids with a matrix bid.

We also show that the winner determination problem of the matrix bid auction is NP-hard even for the special case where all bidders have an identical ranking of the items. We present two formulations for the winner determination problem, and prove that their LP-relaxations are equally strong. One formulation is used to develop two branch-and-price algorithms for which the pricing problem is a shortest path problem. These algorithms are tested on instances with up to 50 items and 100 bidders. All instances can be solved within a reasonable amount of time, and the branch-and-price algorithms withstand the comparison with a state-of-the-art branch-and-cut approach.

Samenvatting

Dit proefschrift richt zich op combinatorische veilingen. Een combinatorische veiling is een veiling waar meerdere goederen tegelijk geveild worden, en waar het de bieders is toegestaan om biedingen uit te brengen op willekeurige deelverzamelingen van de aangeboden goederen. Wanneer meerdere goederen in beschouwing genomen worden, is het vaak het geval dat de waardering voor een verzameling van items hoger of lager is dan de som van de waarderingen voor de individuele items. Dit noemt men respectievelijk een complementariteits- of een substitutie-effect. Deze soms aanzienlijke effecten kunnen verschillen van bieder tot bieder. Een combinatorische veiling laat bieders toe deze effecten in hun biedingen uit te drukken, wat ook de veilingmeester ten goede kan komen. Dit proefschrift is vooral gericht op het zogenaamde *winnaar determinatie probleem*, wat erin bestaat te beslissen welke biedingen aanvaard en welke afgewezen moeten worden zodat de totale waarde van winnende biedingen gemaximaliseerd wordt. Elke winnende bieder betaalt dan hetgeen hij geboden heeft voor de goederen die hij wint. In het algemeen is dit probleem echter NP-lastig en is de optimale oplossing ook moeilijk te benaderen. In de literatuur vindt men vele pogingen om met die complexiteit om te gaan. Eén benadering is om beperkingen op te leggen omtrent hetgeen een bieder kan bieden. Dit proefschrift kan worden onderverdeeld in twee delen, waarin telkens een speciaal geval van een combinatorisch veiling behandeld wordt waar de voorkeuren van de bieders telkens op een bepaalde manier gelimiteerd worden.

Het eerste deel van dit proefschrift handelt over de zogenaamde *veiling met totale hoeveelheidskorting.* In deze veiling zijn de bieders verkopers en komt een bieding overeen met een aanbod om een verzameling van goederen te verkopen tegen een zekere prijs. In opdracht van een koper is het dan de taak van de veilingmeester om een aantal biedingen te aanvaarden zodat opgelegde hoeveelheden van die goederen gekocht kunnen worden tegen een minimale totale kost. Verder verleent elke bieder in zijn biedingen een korting die afhangt van het totale aantal goederen dat de koper aankoopt bij die bieder. Meer specifiek wordt er bij elk bod een volume interval gespecificeerd waarin de totale hoeveelheid goederen (die bij die bieder gekocht wordt) moet liggen, vooraleer het bod aanvaard kan worden. Verder heeft elk bod ook een eenheidsprijs voor elk goed dat door de betreffende bieder wordt aangeboden. Om tot een korting te komen is het dan natuurlijk de bedoeling dat bieders hun prijzen laten zakken in biedingen met een hoger volume interval.

Een theoretisch resultaat bestaat erin dat het winnaar determinatie probleem van deze veiling NP-lastig is. Dit resultaat blijft geldig voor een aantal varianten van dit probleem. Meer zelfs, onze resultaten tonen ook aan dat het bestaan van een algoritme dat een benadering binnen een constante factor garandeert zou impliceren dat $P = NP$. Een ander theoretisch resultaat is dat de LP-relaxatie van een formulering voor dit probleem opgelost kan worden door een minimale kosten stroom probleem op te lossen. We gebruiken dit resultaat om een branch-and-bound (vertak en begrens) algoritme te ontwikkelen dat een minimale kosten stroom probleem oplost in elke knoop van de corresponderende zoekboom. Dit algoritme kan ook gebruikt worden voor vier varianten van de veiling met totale hoeveelheidskorting. In een eerste variant is het marktaandeel dat één of meerdere bieders kunnen behalen beperkt. Een andere variant laat toe dat de koper meer goederen koopt dat oorspronkelijk voorzien, om op die manier een lagere totale kost te bereiken. In een derde variant wordt het aantal winnende bieders gelimiteerd, zowel in het algemeen als per goed. Tenslotte bestuderen we ook een variant die over meerdere perioden loopt en waarin de koper niet alleen moet beslissen welke items hij bij welke bieder koopt, maar ook wanneer hij dit moet doen, rekening houdend met voorraadkosten.

Onze empirische resultaten zijn gebaseerd op experimenten op instanties met tot 50 bieders en 100 goederen. We vergelijken de prestaties van het algoritme dat gebaseerd is op het minimale kosten stroom probleem met twee andere algoritmes: een branch-and-bound algoritme gebaseerd op lineair programmeren en een branch-and-cut (vertak en snij) algoritme. Het blijkt dat zelfs de grote instanties van de basisversie van het probleem optimaal opgelost kunnen worden in een beperkte tijd. Het is bovendien zo dat het beste algoritme in termen van rekentijd afhangt van de variant en de probleemgrootte.

Tot slot van het eerste deel bestuderen we de zogenaamde uitsluitingsbeperkingen. In de veiling met totale hoeveelheidskorting dwingt dit type beperking af dat er ten hoogste één bod per bieder een winning bod kan zijn. We proberen inzicht te krijgen in hoe deze beperkingen bijdragen tot de complexiteit door ze te bestuderen in het kader van een transport probleem. In deze context impliceert een uitsluitingsbeperking op een paar van aanbodknopen dat ten hoogste één van deze knopen gebruikt kan worden om een vraagknoop te bevoorraden. Drie speciale gevallen van het transportprobleem met uitsluitingsbeperkingen worden beschouwd, waarvan er één in het bijzonder goed lijkt op het winnaar determinatie probleem van de veiling met totale hoeveelheidskorting. We bepalen de complexiteit van elk van deze speciale gevallen.

In het tweede deel van dit proefschrift bestuderen we de zogenaamde *matrix bod veiling*. Deze veiling beperkt de bieder in het uitdrukken van zijn voorkeuren in die zin dat de waarde dat een item toevoegt aan een bod op een verzameling van goederen enkel mag afhangen van het aantal hoger gerangschikte goederen in die verzameling. Die rangschikking van de goederen is door de bieder vastgelegd, en kan dus variëren van bieder tot bieder. We tonen aan hoe kan worden nagegaan of een matrix bod voldoet aan economische eigenschappen zoals *free disposal*, *complement freeness*, afnemende marginale waarderingen, en de bruto substituten eigenschap. Verder voorzien we een procedure om na te gaan of een gegeven verzameling biedingen kan voorgesteld worden door een matrix bod. Voor het geval waarin dat niet kan, presenteren we een methode om deze biedingen te benaderen via een matrix bod.

We tonen ook aan dat het winnaar determinatie probleem van de matrix bod veiling nog steeds NP-lastig is in het speciale geval waar elke bieder dezelfde rangschikking van de goederen gebruikt. We presenteren twee formuleringen voor het winnaar determinatie probleem, en bewijzen dat hun LP-relaxaties even sterk zijn. Eén formulering wordt gebruikt om twee branch-and-price (vertak en prijs) algoritmes te ontwikkelen waarvan het pricing probleem een kortste pad probleem is. Deze algoritmes worden getest op instanties met tot 50 goederen en 100 bieders. Alle instanties worden opgelost in een redelijke tijd, en de branch-and-price algoritmes doorstaan de vergelijking met een actuele branch-and-cut aanpak.

Contents

Chapter 1

Introduction

A combinatorial auction is an auction where multiple items are for sale simultaneously to a set of buyers. In a combinatorial auction a buyer is allowed to place bids on subsets of the items. These subsets are sometimes called *bundles*. The auctioneer decides – after one or more rounds or after a certain amount of time depending upon the design – to accept some of the bids and to allocate the items accordingly to the bidders. This thesis mainly deals with this decision: which bids to accept, in order to optimize a given objective. We investigate how this problem for particular settings can be solved.

This introductory chapter is organized in three sections. Section 1.1 provides a general introduction to the realm of combinatorial auctions. In section 1.2, we focus on one of the most challenging problems in this field, namely the winner determination problem. Finally, in section 1.3, we present an outline of the remainder of this thesis.

1.1 Combinatorial auctions

The main advantage of a combinatorial auction is that it allows a bidder to express his[1] preferences to a greater extent. Indeed, it may occur that a bidder values a set of items higher than the sum of his valuations for the individual items of this set. If this is the case, we say that these items are complements to this bidder; if the converse is true we say these items are substitutes. Notice that these complementarity or substitution-effects may be bidder-specific. In a traditional auction, where items are auctioned sequentially, the presence of complementarity or substitution-effects can be problematic for the bidder. Suppose for instance that a bidder is willing to pay a price of €10 for items A and B together. The items individually are worth far less to this bidder, say €2 per item. Suppose this bidder wins the auction for item A, paying €4. The auction for item B however turns out unfavorable, since some other bidder is willing to pay €7, which is more than what our bidder had in mind. Eventually, our bidder ends up paying €4 for item A, which he valued at no more than €2. This problem is known as the *exposure problem*. This problem could also have repercussions for the auctioneer, since in future auctions, our bidder will probably drop out earlier in a similar situation, to limit his loss. Thus, it is important for the auctioneer not to neglect these effects. A combinatorial auction offers a way to make use of complementarity or substitution-effects.

The popularity of combinatorial auctions (and corresponding research) has increased in recent years. As a result, there are many examples where combinatorial auctions prove to be a successful way to market items. The first steps toward implementing a combinatorial auction in practice were taken by the Federal Communications Commission (FCC) when auctioning U.S. spectrum rights. A spectrum right is the right to use a specific bandwidth of the electromagnetic spectrum for wireless communication devices (cell phones, pagers, etc.). Obviously, sets of spectrum rights can be seen

[1] his can be replaced by her (and he by she)

as complements, for instance if they cover geographically adjacent regions. An overview of the considerations in designing the FCC spectrum auctions over the years can be found in Cramton (2002). The airline sector also offers possibilities for combinatorial auctions: landing slots (Ball, Donohue & Hoffman 2005, Rassenti, Smith & Bulfin 1982), seats (Eso 2001) and flights (Bleischwitz & Kliewer 2005). We also find applications of combinatorial auctions in truckload transportation (Caplice & Sheffi 2005, Ledyard, Olson, Porter, Swanson & Torma 2002, Sheffi 2004) and allocating bus routes (Cantillon & Pesendorfer 2005). We conclude this non-exhaustive overview by mentioning Epstein, Henríquez, Catalán, Weintraub & Martínez (2002), who use a combinatorial auction to assign catering contracts for meals in Chilean schools.

Years of research have produced a long list of combinatorial auction mechanisms, containing e.g. the ascending proxy auction (Ausubel & Milgrom 2002), the primal-dual auction (de Vries, Schummer & Vohra in press), *i*Bundle (Parkes & Ungar 2000), PAUSE (Kelly & Steinberg 2000), and the clock-proxy auction (Ausubel & Milgrom 2005). The most famous combinatorial auction is undoubtedly the Vickrey-Clarke-Groves (VCG) auction, which is a generalization of the Vickrey auction to a setting with multiple items. In the Vickrey auction, bidders submit a sealed bid for the item that is being auctioned. The highest bidder wins the item, and pays the amount of the second highest bid. This setting gives each bidder the incentive to bid his true valuation of the item (Vickrey 1961). In the VCG auction, bidders must report a bid function and items are subsequently allocated in order to maximize the total winning bid value. Each winning bidder then pays the opportunity cost of his participation in the auction: the maximal total winning bid value if this bidder had not participated, minus the actual total winning bid value with his winning bid subtracted out. The VCG auction also has the property that each bidder has the incentive to bid his true valuation of the items (see Ausubel & Milgrom (2005)).

Bichler, Kalagnanam, Lee & Lee (2002) outline a classification of allocation problems based on the number of participants and the type of traded goods. Along the former criterion, these authors discern settings with one buyer and one seller (bilateral allocation problem), multiple buyers and sellers (N-bilateral allocation problem), and other involved parties, beside buyers and sellers (multilateral allocation problem). The latter criterion leads to a classification in three dimensions: the number of different items in the auction (single-item versus multi-item), the quantity for each item in the auction (single-unit versus multi-unit) and the negotiable qualitative attributes (single-attribute versus multi-attribute). Bichler et al. (2002) define combinatorial auctions as multi-item auctions. Furthermore, we can make a distinction between forward and reverse auctions. While forward auctions involve a single seller and multiple buyers, a reverse auction (or procurement auction) involves multiple sellers and one buyer. As a typical example, this buyer wants a variety of tasks to be carried out, while sellers are in principle willing to perform subsets of tasks. In such a setting a bid for a subset of tasks is an offer to fulfill that subset of tasks for the stated price. The goal of the auctioneer is to have all tasks carried out at minimal cost.

It will be clear that – compared to a traditional auction where only one item at a time is being auctioned – combinatorial auctions bring about some extra problems. To begin with, there is the question of how bids should be expressed. There is a variety of bidding languages available, and a trade-off between expressiveness and simplicity is to be made here (see e.g. Nisan (2005). Next, the bids need to be communicated to the auctioneer. If each bidder of the n bidders expresses his preferences on all subsets of the m items that are being auctioned (supposing that the bidders would be willing to expose their private information like this), this results in $O(n2^m)$ pieces of information to be submitted to the auctioneer. Bid submission and the relation between the extensiveness of this communication and the allocation are studied in e.g. Nisan & Segal (2006). Further, the problem to decide which bids to accept and which bids to turn down is known as the winner determination problem. This allocation decision depends of course on the auctioneer's objective. In this thesis we assume that the objective is to maximize the total value of the winning bids. Each winning bidder thus pays his bid for the set of items he wins. Notice that in general, the payment of a bidder need not be equal to his bid. We refer for instance to the Vickrey-Clarke-Groves auction. Since the winner determination problem is the main focus of this thesis, a more thorough introduction to this problem is given in section 1.2. Finally, many other auction design questions (how many rounds will be held? what feedback will be given to the bidder? etc.) and incentive issues (will bidders bid their actual valuation? can free rider problems be avoided? etc.) are worth considering.

In this work, we use the private valuation model, introduced by Vickrey (1961). In this model, each bidder can express his valuation for any subset of the items in terms of "money". Moreover, these values do not depend on the private information of other bidders. Furthermore, we assume that there are no externalities present in the valuation of a bidder. This means that the valuation of a bidder for a set depends only on the items in that set. Thus, a bidder does not care to which particular bidder the other items would be allocated. We also assume that bids are normalized, in the sense that any bidder's bid on the empty set is zero.

This thesis relies heavily on important results achieved in various domains. However, these results are often not discussed in great detail in this text. Therefore, we refer to Garey & Johnson (1979) and Ausiello, Crescenzi, Gambosi, Kann, Marchetti-Spaccamela & Protasi (1999) for an excellent overview on fundamental concepts and findings in complexity theory. Naturally, this work is also built on decades of research on auctions in general. A fine introduction to auction theory can be found in Klemperer (2000) and Milgrom (2004). The most fascinating results in operations research and combinatorial optimization can be found in Winston (2004) and Nemhauser & Wolsey (1988). Finally, for a thorough discussion on combinatorial auctions, integrating contributions on many interesting aspects from both theory and practice, we refer to the book edited by Cramton, Steinberg & Shoham (2005).

1.2 The winner determination problem

In a combinatorial auction in its most general form, bidders can bid whatever amount they please on any subset of items in which they are interested. The problem of deciding which bidders should get what items in order to maximize the total winning bid value is called the winner determination problem. Furthermore, we assume that at most one bid per bidder can be accepted. Indeed, a bidder's bid for the whole set of items he wins might well be smaller than the sum of bids for the underlying subsets. Suppose for instance a bidder j expresses the following bids: $b_j(\{1\}) = 3$, $b_j(\{2\}) = 2$, and $b_j(\{1, 2\}) = 4$. Accepting both the bid on item 1 and the bid on item 2 leads to a combined bid of $2 + 3 = 5$, whereas this bidder intended to bid no more than 4 for the combination of items 1 and 2.

The following formulation is most commonly used to represent this problem (for a single-unit setting). It makes use of the following notation, which we use throughout this thesis. We use B to represent the set of bidders indexed by j and G for the set of items indexed by i. We use $b_j(S)$ to denote the bid by bidder j on the set of items $S \in \Omega_j \subseteq 2^G$, where Ω_j is the set of sets in which bidder j is interested. The binary variable $y(S, j)$ indicates whether bidder j wins the set S ($y(S, j) = 1$), or not (otherwise).

maximize

$$\sum_{j \in B} \sum_{S \in \Omega_j} b_j(S) y(S, j) \tag{1.1}$$

subject to

$$\sum_{S\in\Omega_j:S\supseteq\{i\}}\sum_{j\in B} y(S,j) \leqslant 1 \qquad \forall i \in G \tag{1.2}$$

$$\sum_{S\in\Omega_j} y(S,j) \leqslant 1 \qquad \forall j \in B \tag{1.3}$$

$$y(S,j) \in \{0,1\} \qquad \forall S \in \Omega_j, \forall j \in B \tag{1.4}$$

The first set of constraints (1.2) enforces that no item can be auctioned more than once. The second set of constraints (1.3) ensures that there is at most one winning bid per bidder. It is pointed out by de Vries & Vohra (2003) that the latter set of constraints can be dropped, if the bid functions are superadditive for all bidders j (i.e. $b_j(S)+b_j(T) \leqslant b_j(S\cup T)$, with $S, T \in \Omega_j$).

The winner determination problem is shown to be NP-hard, even if every bidder bids only on subsets of size 2 and all bids have a value equal to 1 (Van Hoesel & Müller 2001). This result is based on a reduction from the three-dimensional matching problem (see Spieksma (1999)). Moreover, Sandholm (2002) shows that the winner determination problem cannot be approximated within a ratio of $\max(K^{\epsilon-1}, m^{\epsilon-1/2})$ in polynomial time for any fixed $\epsilon > 0$ (unless $P{=}ZPP$), where K is the number of subsets of G on which a bid has been made, and m is the number of items. This result holds even if every item occurs in at most two bids and all prices are equal to 1.

Numerous attempts to cope with this computational complexity can be found in literature. From the auctioneer's point of view, these attempts can be subdivided in three categories, of which we give a short overview hereunder. A first group of attempts (called *decentralized* approaches) boils down to shifting the burden of solving the winner determination problem (at least partially) to the bidder. In this approach, each time a new bid is expressed, the bidder should also indicate how this bid can lead to an improved allocation. Another way to tackle the complexity consists of restricting the subsets on which bids can be placed. In a third approach, bids can be submitted on all possible subsets, but there are restrictions on what a bidder can bid for these sets. The distinction between restricting the subsets and restricting the preferences is also used in Müller (2005).

1.2.1 Decentralized approaches

In the decentralized approaches, there is no auctioneer who collects the bids from all bidders and then decides which bidder is to get what items. Instead, the auctioneer merely proposes an initial allocation, leaving it up to the bidders to come up with a new allocation that generates a higher total winning bid value (and is obviously also more satisfying for the bidder proposing this allocation).

In the Adaptive User Selection Mechanism (AUSM, see Banks, Ledyard & Porter (1989)) a bidder can submit a bid on a subset, if that bid increases the total winning bid value over the bids it drives out of the current allocation. To this end, the bidder is allowed to incorporate one or more unsuccessful bids from other bidders into his own bid. This is done iteratively, until an allocation is found for which no bidder expresses a higher bid within some amount of time since the last bid was made.

Kelly & Steinberg (2000) present the Progressive Adaptive User Selection Environment (PAUSE) procedure, which consists of two multi-round stages. In the first stage, the items are auctioned individually in simultaneous auctions over a number of rounds. In each round of the second stage, each bidder submits an allocation of all items, incorporating bids of other bidders if necessary. Thus, such a bid suggests not only which items are desired by the submitting bidder, but also to which bidder(s) the other items can be allocated. In both stages, there are requirements on the improvement a new bid creates and there is an activity rule for the bidders.

In a decentralized approach, the auctioneer only needs to check the validity of each bid, which is a computationally tractable problem. However, the burden of complexity has been shifted to the bidder, who now faces the problem of deciding whether he can create a bid that beats the current allocation. Although this problem may still be NP-complete, the idea is that it is easier to solve in practice than the general winner determination problem, because a bidder is often only interested in a limited number of items. Additionally, the bidder no longer needs to reveal his entire valuation function to the auctioneer, which is often beneficial for privacy reasons.

1.2.2 Restricted-subset approaches

Unlike the decentralized approaches, the restricted-subset approaches stick to a combinatorial auction where the auctioneer is solely responsible for finding an optimal allocation. In order to reduce the complexity of the winner determination problem, bids can be submitted only on a limited set of combinations of items.

Rothkopf, Pekeç & Harstad (1998) found that if the family of subsets on which a bidder can bid is limited to hierarchical subsets, meaning that every two subsets are disjoint or one is a subset of the other, that then the winner determination problem can be solved in polynomial time. The problem of finding an optimal allocation for a combinatorial auction where a linear order exists among the items and bidders can only bid on subsets of consecutive items, is also shown to be polynomially solvable. If the first item in the ordering is considered the successor of the last (i.e. a circular order), then the winner determination problem remains solvable in polynomial time. Furthermore, Rothkopf et al. (1998) prove that a combinatorial auction where bidders can bid on subsets of a cardinality of at most two has a polynomially solvable winner determination problem. However, the results of Rothkopf et al. (1998) assume a combinatorial auction where a bidder is allowed to win multiple subsets.

Nisan (2000) elaborates on some of the results of Rothkopf et al. (1998) by stating explicitly that the LP-relaxation of a set packing formulation (i.e. (1.1)-(1.4), without constraints (1.3)) results in an integral solution for the special cases with hierarchical subsets and linearly ordered items. These results follow from the fact that a set packing formulation with a totally unimodular constraint matrix defines a polyhedron where all extreme points are integral (see Nemhauser & Wolsey (1988)).

Tennenholtz (2002) presents a combinatorial network auction, which he proves is computationally tractable. In this auction, the items are assumed to be arranged in a tree, where every node corresponds to an item. The idea is that bids can be submitted only on subsets of items that form a path in the network. When the underlying tree is a path, this auction reduces to the combinatorial auction with linearly ordered items (see Rothkopf et al. (1998)). If the items are structured in a directed acyclic graph and the bids are allowed on any directed subtree, the winner determination problem already becomes NP-hard again (Sandholm 2002).

In the above approaches, the auctioneer decides in advance on which combinations the bidders are allowed to bid. While bidders may sometimes agree on what combinations are interesting from an economical point of view, it may also be the case that some bidders experience synergies between some kind of items, while others have totally different kinds of synergies. Obviously, in general this approach can lead to inefficiencies, because bidders may not be allowed to bid on the combinations they want. The task of deciding to which subsets the auction is restricted can therefore be tricky for the auctioneer, especially given the fact that the appearance of fairness should be upheld. Park & Rothkopf (2005) try to avoid this difficulty by proposing an auction in which the bidders themselves determine the allowable combinations. While in principle, bids on all subsets are allowed, the bidders are to assign priorities to various combinations and then may use as many of these (from the highest priority first) as is computationally manageable. Although limiting the number of bids a bidder can express does not make the winner determination problem polynomially solvable (see Rothkopf et al. (1998)), it does increase the size of problems that can be solved within a reasonable amount of time.

Günlük, Ladányi & de Vries (2005) describe an auction that is comparable to the auction of Park & Rothkopf (2005), since in each round, the bidders can submit bids on any single item and on a limited number of combinations of items, which they can compose themselves. For each bidder, a pair of bids from a different round, as well as a pair of bids from a same round that overlap, cannot both be awarded. The authors present a branch-and-price algorithm based on a multi-unit adaptation of formulation (1.1)-(1.4). In their approach, the pricing problem is a vertex packing problem. Although in general this problem is NP-hard, the pricing problem can be solved in a reasonable amount of time for the FCC spectrum auctions. This particular auction is a simultaneous ascending auction with package bidding, where bidders compete for licenses for spectrum use (e.g. radio, wireless telephone, etc.). Since the bidders are not allowed to submit too many package bids in any round, the pricing problem remains computationally manageable, allowing the branch-and-price algorithm to solve instances with up to 99 items within 10 minutes.

1.2.3 Restricted-preference approaches

Instead of limiting the sets for which a bid can be placed, a restricted-preference method imposes limitations to what a bidder is allowed to bid on a bundle. A common restriction on a bidder's preferences is that they should be non-decreasing, i.e. the valuation for a set S_1 can not be higher than the valuation for a set S_2 if S_1 is a subset of S_2. Another restriction can be that the preferences should be supermodular. This means that the sum of valuations for two sets should not be higher than the sum of the valuation of the union of both sets and the valuation of the intersection of both sets. If there are only two bid functions a bidder can have, both of them non-decreasing, integer valued and supermodular, then de Vries & Vohra (2003) point out that the winner determination problem of this auction can be solved in polynomial time. The authors also mention that if each bidder's bid function satisfies the gross substitutes property (see Kelso & Crawford (1982) and section 6.3.4), the LP-relaxation of (1.1)-(1.4) provides an integral optimal solution.

If a bidder has a sequence of valuations $p_1 \geq p_2 \geq ... \geq p_m$, where p_j specifies his valuation for the j-th item he wins, regardless of what that item might be, he has a so-called downward sloping symmetric bid function (Nisan 2000). Notice that this means that the bidder values all items as if they are identical and the bid is therefore only determined by the number of items in the set. Nisan (2000) proves that if all bidders have a bid function of this kind, the LP-relaxation of (1.1)-(1.4), without constraints (1.3) yields an integral optimal solution. Notice that this assumes that bidders are allowed to win multiple bids. A more general structure is captured in so-called matrix bids with order, which are intensively investigated in chapter 6.

Tennenholtz (2002) presents a quantity-constrained auction where all bids consist of a series of desired items, each with a price the bidder is willing to pay for this item, and a maximal number of items that he wishes to win. Notice that this is a subadditive valuation function, since a bidder will not pay extra for anything exceeding his quantity limit. Tennenholtz (2002) shows that the winner determination problem for this auction is computationally tractable. Furthermore, this author presents a quantity-constrained auction with binary combinatorial bundles and an almost-additive auction, which he proves also to be computationally tractable.

1.3 Outline of the thesis

The remainder of this thesis consists of two parts, in each of which we investigate a combinatorial auction in which bidder's preferences are restricted in some way. In chapters 2 to 5, we study a reverse auction, which we call the *total quantity discount auction*. A forward auction, the so-called *matrix bid auction*, is the topic of study in chapters 6 and 7.

In chapter 2, we study a procurement problem where a buyer needs to purchase multiple units of different items at a minimal total cost. The suppliers use a discount policy based on the total amount of units the buyer purchases. We show how this problem can be solved by means of an auction, namely the total quantity discount auction. Despite the fact that we prove the resulting winner determination problem to be NP-hard and difficult to approximate, we find that this problem has an interesting property with respect to solving it efficiently.

Four variants of the total quantity discount auction are discussed in chapter 3. In a first variant, the market share that one or more suppliers can obtain is constrained. Another variant allows the buyer to procure more units than strictly needed, in order to reach a lower total cost. We also consider a setting where the buyer needs to pay a disposal cost for the extra units bought. In a third variant, the number of winning suppliers is limited, both in general and per product. Finally, we investigate a multi-period variant, where the buyer not only needs to decide what items to buy from what supplier, but also when to do this, while considering the inventory costs.

In chapter 4, we discuss three exact algorithms (min-cost flow based branch-

and-bound, linear programming based branch-and-bound, and branch-and-cut) and investigate their performance on randomly generated instances involving 50 suppliers and 100 items. It turns out that even the large instances of the basic problem are solved to optimality within a limited amount of time. However, we find that different algorithms perform best in terms of computation time for different variants. The results described in chapters 2, 3, and 4 form the basis of a paper to appear in EJOR (Goossens, Maas, Spieksma & van de Klundert 2007).

Chapter 5 can be seen as an interlude in this thesis. In this chapter, we take a closer look at the so-called *exclusionary side constraints*. In the total quantity discount auction, this type of constraint enforces that only one bid per supplier can be the winning bid. We try to gain insight in how these constraints add to the complexity by studying it in the context of the transportation problem. Three special cases of the transportation problem with exclusionary side constraints are considered, of which one particularly resembles the total quantity discount auction. A research report is based on this chapter (Goossens & Spieksma 2005).

In chapter 6, we study the matrix bid auction. This auction limits a bidder in expressing his preferences, in the sense that the extra value an item adds to a bid on a set is determined only by the number of items in this set outranking this item, according to an ordering of the items expressed by the bidder. We show how it can be verified whether a matrix bid satisfies economic properties as free disposal, complement freeness, decreasing marginal valuations and the gross substitutes property. Finally, we provide a procedure to check whether a given collection of bids can be represented by a matrix bid. If this is not the case, we present a method to approximate these bids with a matrix bid. This chapter has been published as a research report (Goossens & Spieksma 2006b).

In chapter 7, we present and compare two formulations for the winner determination problem of the matrix bid auction. To solve this problem, we develop two branch-and-price algorithms for which the pricing problem is a shortest path problem. These algorithms are tested on randomly generated instances with up to 50 items and 100 bidders. The chapter concludes with a discussion of the computational results. The findings discussed in this chapter have also been published as a research report (Goossens & Spieksma 2006a).

To conclude this thesis, we elaborate on a number of topics for future research in chapter 8.

Chapter 2

The total quantity discount auction

It is a widespread economic phenomenon that the price of an item depends – among many other things – on the amount ordered. Indeed, there are many reasons for suppliers to offer discounts based on the volume sold to a buyer. Consequently, when it comes to procuring amounts of different items from different suppliers, it makes sense to consider various alternatives. In fact, choosing the right suppliers to deliver the right products has become a major concern in many large companies. In this chapter, we investigate a combinatorial auction tailored to solve a procurement problem where bidders use a specific discount policy. We call this auction the *total quantity discount auction*. In section 2.1, we describe the discount structure and illustrate how this auction works. In section 2.2 we motivate why this discount policy is relevant by providing a number of real-life examples. Section 2.3 gives an overview of related research on procurement and discount structures. A mathematical formulation for the winner determination problem of the total quantity discount auction is given in section 2.4, while in section 2.5, we discuss a number of interesting properties of this problem.

2.1 Description of the total quantity discount auction

The total quantity discount auction is a reverse auction, meaning that there is one buyer that needs a number of items and several suppliers compete for the right to provide items. The total quantity discount auction is also a multi-item, multi-unit auction, since the buyer can require many different items, and also multiple units of each item. The auction is cleared in a single round and the outcome determines which bidders are allowed to supply what items such that the buyer pays a minimal total amount. In this way, the auction solves the procurement problem faced by the buyer.

Let's assume that the buyer is interested in a number of items, and has a quantity for each of these items that he needs to purchase, from one or more suppliers. To compete for (a part of) this request, each bidder (or supplier) needs to submit one or more bids, consisting of a lower and and upper bound of a volume interval, and prices for each item the supplier is willing to offer. The idea is that this bid is only valid if the buyer purchases a total amount of units that lies within the ranges of the volume interval.

Naturally, a bidder can submit multiple bids, however, we assume that the corresponding volume intervals do not overlap. By submitting multiple bids, the supplier can offer a discount that depends on the total quantity the buyer orders from that supplier. Obviously, a supplier is assumed not to increase his prices in a higher volume interval. Thus, it is the total amount of units (possibly of various items) that determines the discount a supplier is willing to grant. Notice however that the discount rate in each interval might vary among the items. This discount structure is called total quantity discount (TQD). Notice that to reach the quantity the buyer requires for an item, the buyer may purchase from multiple suppliers, although he can accept only one bid per supplier. Clearly, this is an all-unit discount policy, since the prices in a volume interval apply to all units bought from the corresponding supplier (as opposed to an incremental discount policy, where only the items purchased in surplus of some threshold give rise to a discount). A discussion and classification of various quantity discount policies can be found in Munson & Rosenblatt (1998).

We assume that a preselection of suppliers has been made, excluding those suppliers who do not attain the required standards with respect to quality, reliability and other relevant considerations from the auction (see Degraeve, Labro & Roodhooft (2000) for a discussion of these considerations). Thus, we assume that the only remaining criterion upon which the further supplier selection decision is based, is the price these suppliers charge for the various items. Given a demand for each item, the winner determination problem for the total quantity discount auction is to determine which items should be purchased from what supplier in order to satisfy the demand for each item at minimal total cost for the buyer. We will refer to this problem as the TQD problem. In the collection of winning bids that supports this solution, at most one bid of each bidder can be present. Obviously, for each winning bid, the constraints on the lower and upper bound of the total volume purchased from the corresponding supplier should be respected.

As an example, we consider a buyer who is throwing a party and wishes to purchase pizza. The pizzas come in three kinds: pizza hawai (H), pizza al ovo (AO), and pizza tirolese (T); the buyer needs four of each kind. There are two pizza delivery companies, "Mamma Mia" and "Mangia e via", who express the following bids.

Mangia e via:

$< [1,5]; p_H = €9, p_{AO} = €10, p_T = €8 >$

$< [6,12]; p_H = €8, p_{AO} = €8.5, p_T = €7 >$

Mamma Mia:

$< [1,12]; p_H = €7, p_{AO} = €8, p_T = €9 >$

If the buyer orders up to 5 pizzas, “Mangia e via” will charge €9 for a pizza hawai, €10 for a pizza al ovo and €8 for a pizza tirolese. However, if 6 pizzas or more are ordered, the buyer can make use of the second bid, where prices drop to €8, €8.5 and €7 respectively. “Mamma Mia” has made only one bid, charging €7 for a pizza hawai, €8 for a pizza al ovo and €9 for a pizza tirolese. The optimal solution for the buyer would be to accept the bid by “Mamma Mia” and the second bid by “Mangia e via”. This would allow him to order 2 pizzas al ovo and 4 pizzas tirolese from “Mangia e via” and 4 pizzas hawai and the other 2 pizzas al ovo from “Mamma Mia”, at a total cost of €89.

The total quantity discount policy is a way to price a set of items: the cardinality of the set of all items ordered determines in which interval the buyer is, and the all-unit discount policy leads to prices that imply complementary effects. Therefore, the total quantity discount auction is indeed a combinatorial auction.

2.2 Motivation

One of the main advantages of the total quantity discount auction is that its winner determination problem has a special structure that can be exploited when constructing an exact algorithm (see section 2.5 and chapter 4). On the other hand, the TQD auction restricts the preferences that can be expressed by a bidder. To illustrate that this restriction can be quite natural in procurement settings, we now describe three recent examples, documented in literature, where suppliers set their prices using a total quantity discount policy. These cases are from the dairy, the chemical, and the telecommunication industry.

A procurement problem where a dairy producer needs to purchase a number of bull semen straws from one or more breeding companies in order to inseminate (a part of) its herd is discussed by McConnel & Galligan (2004). Each breeding company offers semen straws from a number of bulls, and grants a volume discount, based on the total amount of semen straws the dairy producer purchases from that breeding company. Furthermore, if a volume interval is reached, the corresponding discounted prices apply to all semen straws purchased from that breeding company. Thus, the breeding companies apply a total quantity discount policy. Each available bull has a score on a number of traits that are of importance to the producer. The number of semen straws needed from each particular bull is however not known in advance. The diary producer only specifies a total number of semen straws needed and a number of trait goals that should be satisfied by the portfolio of semen straws on average. Furthermore, in order to reduce the risk of inbreeding, there is a constraint that limits the purchased amount of semen straws from any bull to a given percentage of the portfolio. McConnel & Galligan (2004) develop a mixed integer program, which also takes into account the possibility to order more straws than needed in order to reach a higher discount interval. The authors use this formulation to solve a real-life instance involving a demand of 1500 semen straws to be satisfied from 52 available bulls across three breeding companies. These companies specify no more than three volume intervals.

Crama, Pascual & Torres (2004) investigate another procurement problem, characterized by a total quantity discount policy. In this problem, a chemical company needs to purchase a number of ingredients from one or more suppliers. Also in this case, the suppliers express the discount as a function of the total quantity of ingredients purchased. Since only one single discount rate for all ingredients is used, this policy should be considered as a special case of the total quantity discount policy used in our problem. Crama et al. (2004) however face the additional problem of deciding how to use the purchased ingredients to manufacture the desired quantities of the endproducts, since there are several alternative recipes, each with different ingredient requirements to produce an endproduct. The authors present a number of mixed integer models which they solve by branch-and-bound, in order to come to a solution for a real-life case where 25 different ingredients can be purchased from 8 suppliers in order to manufacture more than 30 distinct products.

A procurement problem in the telecom industry is described by van de Klundert, Kuipers, Spieksma & Winkels (2005). Consider a telecommunication company that needs to acquire capacity to accommodate its international calls. This capacity is offered by various so-called carriers, i.e., for each destination, each carrier offers capacity, priced in eurocents per minute. Prices of carriers differ, and – which is particularly relevant for our setting – each carrier uses an interval structure to arrive at a certain price. In other words, the total amount of call-minutes handled by a certain carrier determines the price. Moreover, the carriers use all-unit discounts. The problem is to acquire the right amount of capacity for each destination at minimal cost. A solution approach based on explicit enumeration of all interval selections is presented by van de Klundert et al. (2005) and used to solve instances with 5 carriers and up to 5000 destinations.

2.3 Related work on discounts and procurement

Procurement problems involving discount policies have been studied by many authors. Katz, Sadrian & Tendick (1994), and also Sadrian & Yoon (1994) discuss a procurement problem where they distinguish between purchases on a commitment basis and purchases on an as-ordered basis. They stress the importance of sourcing flexibility and model explicitly the fact that not all future items should be purchased via committed contracts. In addition, they explicitly consider the number of vendors for each item, and the percentages of the total supply given to each of the vendors. In their discount policy, a supplier discounts the price of each item by the same percentage based on the total dollar value of all items purchased from the supplier, whereas our policy allows a different discount percentage for each item.

Austin & Hogan (1976) is an early reference to procurement problems characterized by a lower and upper bound for each supplier between which the ordered amount needs to lie, provided that that supplier is used. In this paper, the government needs to purchase a given amount of aviation fuel from one or more suppliers, where prices differ depending on how the fuel is transported. This problem differs from our setting in that the items considered are independent and there are no discounts. The authors solve the problem using a branch-and-bound algorithm, exploiting the network structure of the core problem.

Davenport & Kalagnanam (2002) report on a volume discount auction in which discounts are based on quantities for each individual item. Furthermore, they use an incremental discount policy, meaning that the discounts apply only to the additional units above the threshold of the volume interval. Hohner, Rich, Ng, Reid, Davenport, Kalagnanam, Lee & Chae (2003) describe a web-based implementation of this procurement auction at Mars Incorporated.

Eso, Ghosh, Kalagnanam & Ladányi (2005) also elaborate on the work of Davenport & Kalagnanam (2002). They study a volume discount auction with piece-wise linear supply curves, allowing discontinuities and all-unit discounts. However, they do require additive separable supply curves, which boils down to assuming that the prices charged by a supplier for different commodities are independent. This makes their problem not truly combinatorial, since synergies or substitutability between different items cannot be reflected in the total price charged by the suppliers. As a result, a total quantity discount structure is not possible in their setting. The authors formulate a column generation based heuristic that provides near-optimal solutions to the bid evaluation problem.

Another procurement auction with marginal decreasing piecewise-constant supply curves is described in Kothari, Parkes & Suri (2003). This auction also allows all-unit discounts, but it deals only with a single item. Kothari et al. (2003) present fully polynomial-time approximation schemes for the winner determination problem and the computation of the corresponding payments of this auction.

Kim & Pardalos (2001) made a study of piecewise linear network flow problems, in which they identify a number of categories and for which they show how they can be transformed to fixed charge network flow problems. In particular, the category with sawtooth arc cost functions is of interest to our research, since it contains the TQD problem with a single item. However, the authors do not consider a setting that could allow for total quantity discounts over multiple items.

The TQD problem is also related to the so-called deal splitting problem introduced by Shachnai, Shmueli & Sayegh (2004). In this problem, a buyer needs to split an order of multiple units from a set of heterogeneous items among a set of sellers, each having bounded amounts of the items, so as to minimize the total cost of the deal. Two variants of the deal splitting problem can be discerned, depending on whether the seller offers packages containing combinations of the items or whether the buyer can generate such combinations using seller-specified price tables. Shachnai et al. (2004) show that for both variants an exact solution can be found in pseudo-polynomial time if the number of heterogeneous items is fixed. Moreover, they develop polynomial-time approximation schemes for several subclasses of instances of practical interest.

2.4 Mathematical formulation

To state a mathematical formulation of the winner determination problem of the TQD auction, we use the following notation. We define G as the set of m items, indexed by i, and B as the set of n suppliers, indexed by j. For each item i in G, we define d_i as the amount of item i to be procured. Each supplier j in B expresses max_j bids, which we index by k. Since each bid defines exactly one volume interval, we can also use the index k to denote the volume interval. In this way, for each supplier $j \in B$, l_{jk} and u_{jk} define the minimum and maximum number of items respectively that needs to be ordered according to bid k by supplier j. For simplicity, we assume that the bids are ordered by increasing lower bound of the corresponding volume interval. In this way, we have a sequence of intervals $Z_j = \{1, ..., max_j\}$, indexed by k, with $l_{jk} < l_{jk'}$ if $k < k'$. Finally, for each supplier $j \in B$, for each interval $k \in Z_j$ and each item $i \in G$, let c_{ijk} be the price for one unit of item i purchased from supplier j according to his k-th bid.

We assume that these parameters satisfy the following assumptions:

$$\forall j \in B, k \neq k' \in Z_j : [l_{jk}, u_{jk}) \cap [l_{jk'}, u_{jk'}) = \varnothing, \tag{2.1}$$

$$\forall j \in B, k \in Z_j \setminus \{max_j\}, i \in G : c_{ijk} \geqslant c_{i,j,k+1}, \tag{2.2}$$

$$\forall j \in B, k \in Z_j, i \in G : c_{ijk} \geqslant 0,\ l_{jk} \geqslant 0,\ u_{jk} \geqslant 0,\ d_i \geqslant 0. \tag{2.3}$$

Assumption (2.1) states that bids by the same supplier should not have overlapping intervals. The requirement that prices should not increase from one interval to the next is expressed in the second assumption. The last assumption reflects that all prices and all quantities ordered are non-negative.

We define the decision variable x_{ijk} as the amount of item i purchased from supplier j according to the prices stated in his k-th bid. Further, we define a binary decision variable y_{jk} which is 1 if bid k is selected for supplier j and 0 otherwise. This leads to the following formulation of the TQD problem, referred to as TQDF.

minimize

$$\sum_{i \in G} \sum_{j \in B} \sum_{k \in Z_j} c_{ijk} x_{ijk} \tag{2.4}$$

subject to

$$\sum_{j \in B} \sum_{k \in Z_j} x_{ijk} = d_i \qquad \forall i \in G \tag{2.5}$$

$$\sum_{k \in Z_j} y_{jk} \leqslant 1 \qquad \forall j \in B \tag{2.6}$$

$$\sum_{i \in G} x_{ijk} - y_{jk} l_{jk} \geqslant 0 \qquad \forall j \in B, k \in Z_j \tag{2.7}$$

$$\sum_{i \in G} x_{ijk} - y_{jk} u_{jk} \leqslant 0 \qquad \forall j \in B, k \in Z_j \tag{2.8}$$

$$x_{ijk} \geqslant 0 \qquad \forall i \in G, j \in B, k \in Z_j \tag{2.9}$$

$$y_{jk} \in \{0, 1\} \qquad \forall j \in B, k \in Z_j \tag{2.10}$$

The objective function (2.4) states that the amount of items i ordered from supplier j according to bid k, times the corresponding price must be minimal. Constraints (2.5) make sure that the demand for each item is met, while constraints (2.6) guarantee that at most one bid per supplier is selected. Constraints (2.7) and (2.8) ensure that if a bid k is selected as a winning bid, the total amount of units purchased from supplier j is between the bounds of the corresponding interval. If bid k is not selected, these constraints ensure that $x_{ijk} = 0$. Constraints (2.9) state that only a non-negative amount can be purchased, while constraints (2.10) define y_{jk} as a boolean variable. Notice that this formulation allows to order nothing from a supplier. Notice also that we do not require integrality of the x-variables; if the demands and the lower and upper bounds of each volume interval are integral, then, assuming the existence of a feasible solution, there always exists an optimal solution of TQDF with integral x-values (see section 2.5).

Let us now discuss how this formulation relates to known classes of integer programming formulations. The TQD problem is related to fixed charge network flow problems (see Nemhauser & Wolsey (1988)). In fact, when omitting constraints (2.6) from the formulation above, the resulting problem can be formulated as a (special) fixed charge network flow problem. Indeed, when one builds a network involving a source with supply $\sum d_i$, a 'demand' node for each item i with demand d_i, and an 'interval' node for each interval of each supplier, the variable x_{ijk} in the formulation above represents nothing else but the flow on the arc from an 'interval' node to a 'demand' node. In particular, this implies that inequalities that are valid for this formulation of the fixed charge network flow problem are also valid for TQDF. However, due to the presence of constraints (2.6), the TQDF formulation is more general than a fixed charge network flow problem. Notice, though, that in the objective function (2.4), there is no fixed cost associated to choosing some interval of some supplier, i.e., in terms of the fixed charge network flow problem, the fixed cost of using an arc is 0.

Finally, one can view the TQD problem as a direct generalization of the ordinary, well-known, transportation problem: given a set of demand nodes, each with demand d_i, given a set of supply nodes each with a supply between a given lower bound l_k and upper bound u_k, given costs per item for each combination of demand node and supply node, and finally, given a collection of subsets of the supply nodes such that at most one node of each subset is allowed to supply a positive amount, find a solution of minimum cost. TQD belongs to this class of generalized transportation problems; as far as we are aware, this problem has not been investigated before. Sun (2002) studies a special case of this generalized transportation problem where for each demand node i, pairs of supply nodes are given such that at most one supply node of each pair is allowed to supply demand node i. These constraints are called exclusionary constraints and form the topic studied in chapter 5.

2.5 Properties of the TQD problem

In this section we establish the complexity of the TQD problem (section 2.5.1). We also show that the LP-relaxation of TQDF can be solved by solving a min-cost flow problem (section 2.5.2).

2.5.1 On the complexity of the TQD problem

We show that the TQD problem is a hard problem to solve when aiming for optimal solutions.

Theorem 1. *The decision version of the TQD problem is strongly NP-complete.*

Proof. We define TQD' as the decision version of the TQD problem, where the question is whether it is possible to buy the required items at a given total purchasing cost K. Obviously, TQD' is in NP, since given a solution it suffices to check the constraints and the value of the solution, which can easily be done in polynomial time. The reduction is from the 3-dimensional matching (3DM) problem.

The decision version of the 3DM problem is described as follows: given a set $M \subseteq X \times Y \times Z$ of triples, where each of the sets X, Y and Z has exactly q elements, is there a matching in M that contains q triples? Every instance of 3DM can be reduced to a TQD' instance in polynomial time. Suppose that the $3q$ elements of the sets X, Y, and Z correspond to $3q$ items and that each 3-element subset in M corresponds to a supplier, so $n = q$ and $m = 3q$. Each supplier has 2 bids. The first bid has an interval with a lower bound of 0 and an upper bound of 2. The price of each item in this bid is 1.

The interval of the second bid has a lower bound of 3 and an upper bound of ∞. The price of each item in this bid is also 1, except for the three items in the 3-element subset corresponding to the supplier, each of which have a price of 0. Each item needs to be purchased exactly once, i.e., $d_i = 1$, $\forall i$. The question is whether the TQD' problem can be solved with a total purchasing cost of 0.

Further, every yes-instance of 3DM corresponds to a yes-instance of TQD'. A solution of 3DM consists of q 3-element subsets, corresponding to q suppliers in the TQD' problem. Purchasing from each of these suppliers exactly the 3 items represented by the 3-element subset enables us to reach every supplier's second interval, where these 3 items can be bought at price 0. Since every element of $X \cup Y \cup Z$ occurs exactly once in the solution of 3DM, every item will also be purchased exactly once in the TQD' solution. Therefore, if 3DM has a solution, it can easily be transformed to a solution of TQD'.

Vice versa, every yes-instance of TQD' also corresponds to a yes-instance of 3DM. A solution of the TQD' problem consists of a number of selected suppliers, together providing every item exactly once at a total cost of 0. If a supplier would provide less than 3 items, the quorum to get in the second interval would not be met, so the cost would not be 0. If the supplier would provide more, the cost would also be strictly positive, because all but these 3 items still have a price of 1 in the second interval. Providing more than one of the 0-priced items would violate the demand constraint stating that each item is to be supplied exactly once. Therefore every selected supplier provides precisely 3 items, namely those that have a price of 0 in the second interval and since $3q$ items need to be provided, q suppliers must be selected. Therefore, for each of the q suppliers selected in the solution of the TQD' problem, there is a corresponding 3-element set in M. Moreover, these q triples define a matching, since every item is bought exactly once. As a consequence, the decision version of the TQD problem is strongly NP-

complete. □

In fact, we can also make the following statement on the approximability of the TQD problem:

Theorem 2. *No polynomial-time approximation algorithm with constant worst-case ratio exists for the TQD problem (unless $P = NP$).*

Proof. Assume that a ρ-approximation algorithm for the TQD problem exists. Consider now an instance of 3DM with $M \subseteq X \times Y \times Z$, and let us build an instance of the TQD problem as in the proof of Theorem 1 with a price of $\rho+1$ for any item bought in the first interval, or bought in the second interval when not belonging to one of the three items of that supplier. Observe that this instance of the TQD problem either has an optimal solution with cost 0 (namely when the 3DM-instance has a matching), or it has an optimal solution with cost at least $\rho + 1$ (when there is no matching in the 3DM instance). Thus, if there is a 3DM-matching the ρ-approximation algorithm must return a zero-cost solution, which contradicts the NP-hardness of 3DM. Hence such an algorithm cannot exist unless $P = NP$. □

Consider the following special case of the TQD problem, where the prices stated in each bid are determined by the prices in the bid with the lowest interval and a common discount rate δ. This discount rate δ determines the price $c_{i,j,k}$ of item i in bid k as a function of the price in bid 1 as follows:

$$c_{ijk} = (1 - \delta)^{k-1} c_{i,j,1} \qquad \forall i, j \text{ and } \forall k > 1 \tag{2.11}$$

We claim that this special case of the TQD problem is still a hard problem.

Theorem 3. *The decision version of the TQD problem with a common discount rate δ is strongly NP-complete.*

Proof. In order to show this problem is NP-complete, we modify the reduction used in Theorem 1 as follows. As in Theorem 1, each supplier has 2 intervals, the first interval ranges from 0 to 2 items, the second from 3 to an unlimited amount of items. The prices of all items in the first interval are 2, except for the three items in the 3-element subset corresponding to the supplier, each of which have a price of 1. Since all suppliers use a common discount rate δ, the prices in the second interval are $(1 - \delta)$ for the three items in the 3-element subset, and $2(1 - \delta)$ for the other items. Each item still needs to be purchased exactly once. The question is now whether this TQD problem can be solved with a total purchasing cost of $m(1 - \delta)$. The same reasoning as in Theorem 1 can be applied to verify that every yes-instance of 3DM corresponds to a yes-instance of the TQD problem with common discount rate and vice versa and that indeed the decision version of the TQD problem with a common discount rate is strongly NP-complete. □

Finally, consider the variant of the TQD problem where the amounts purchased must be *at least as large* as the demands d_i. In such a setting, it might happen that buying more than what is strictly needed reduces the total cost. We refer to this problem as the more-for-less variant of the TQD problem (see section 3.2). For the special case of this variant where only one item needs to be purchased, Chauhan, Eremeev, Romanova, Servakh & Woeginger (2005) show that there exists a fully polynomial-time approximation scheme. We claim that this variant remains a hard problem.

Theorem 4. *The decision version of the more-for-less variant of TQD problem is strongly NP-complete.*

Proof. In the more-for-less setting, the buyer is allowed to purchase more than d_i units of any item i in order to reduce the total cost. We can however use the same reduction as in Theorem 3. Indeed, let each supplier have 2 bids, the first with an interval ranging from 0 to 2 items, the second with an interval from 3 to an unlimited amount of items. Once again, the prices of all items in both the first and second interval are 1, except for the three items in the 3-element subset corresponding to the supplier, each of which have a price of $(1-\delta)$. The question remains whether it is possible to solve this TQD problem with a total purchasing cost of $m(1-\delta)$. Clearly this can not be achieved by purchasing more than m units, which allows us to conclude that every yes-instance of 3DM corresponds to a yes-instance of the more-for-less variant and vice versa. Hence, the decision version of the more-for-less variant of the TQD problem is strongly NP-complete. □

2.5.2 Min-cost flow and the winner determination problem

We now show that the LP-relaxation of TQDF can be solved by solving a min-cost flow problem. In fact, even in the more general case where for some suppliers the winning bids are prespecified, the LP-relaxation of the resulting model can still be found by solving a min-cost flow problem. We will use this result to construct an exact algorithm in chapter 4.

Let us first state a model which assumes that for an arbitrary given subset of suppliers, referred to as D ($D \subseteq B$), a bid and its corresponding interval, say $s(j) \in Z_j$, has been selected, while for the remaining suppliers no bid has been selected. We refer to the following formulation as GENTQDF.

minimize

$$\sum_{i \in G} \sum_{j \in B} \sum_{k \in Z_j} c_{ijk} x_{ijk} \tag{2.12}$$

subject to

$$\sum_{j \in B} \sum_{k \in Z_j} x_{ijk} = d_i \qquad \forall i \in G \tag{2.13}$$

$$\sum_{k \in Z_j} y_{jk} \leqslant 1 \qquad \forall j \in B \setminus D \tag{2.14}$$

$$\sum_{i \in G} x_{ijk} - y_{jk} l_{jk} \geqslant 0 \qquad \forall j \in B \setminus D, k \in Z_j \tag{2.15}$$

$$\sum_{i \in G} x_{ijk} - y_{jk} u_{jk} \leqslant 0 \qquad \forall j \in B \setminus D, k \in Z_j \tag{2.16}$$

$$\sum_{i \in G} x_{i,j,s(j)} - l_{j,s(j)} \geqslant 0 \qquad \forall j \in D \tag{2.17}$$

$$\sum_{i \in G} x_{i,j,s(j)} - u_{j,s(j)} \leqslant 0 \qquad \forall j \in D \tag{2.18}$$

$$x_{ijk} \geqslant 0 \qquad \forall i \in G, j \in B \setminus D, k \in Z_j \tag{2.19}$$

$$x_{i,j,s(j)} \geqslant 0 \qquad \forall i \in G, j \in D \tag{2.20}$$

$$x_{ijk} = 0 \qquad \forall i \in G, j \in D, k \neq s(j) \tag{2.21}$$

$$0 \leqslant y_{jk} \leqslant 1 \qquad \forall j \in B \setminus D, k \in Z_j \tag{2.22}$$

Observe that if $D = \varnothing$, the resulting model is the LP-relaxation of TQDF, whereas if $D = B$, we arrive at the situation where an interval has been selected for each supplier (see van de Klundert et al. (2005)). Introducing D allows us to develop an enumerative algorithm solving only min-cost flow problems (see chapter 4).

Theorem 5. *GENTQDF can be polynomially transformed to min-cost flow.*

Proof. We organize the proof by first showing that an optimal solution of GENTQDF has a structural property. Then we construct a min-cost flow instance and show the correspondence between optimal solutions of this instance and GENTQDF.

Claim: There exists an optimal solution (x^*, y^*) of GENTQDF in which for each $j \in B \setminus D$:

$$\begin{aligned} x^*_{ijk} &= 0 \quad \forall i \in G, \forall k \neq max_j, \text{ and} \\ y^*_{ij} &= 0 \quad \forall k \neq max_j. \end{aligned} \tag{2.23}$$

Thus, the claim states that there exists an optimal solution in which all x-

and y-variables equal 0, except those corresponding to the highest interval of each supplier. In other words, items are bought only at the lowest prices of each supplier.

Argument: given some feasible solution (x, y) of GENTQDF, we show how to modify (x, y) to (x^*, y^*) such that (x^*, y^*) is a feasible solution of GENTQDF satisfying (2.23) and such that the cost of (x^*, y^*) does not exceed the cost of (x, y).

For each $i \in G$ and each $j \in B \setminus D$, we set

$$x^*_{i,j,max_j} = \sum_{k=1}^{max_j} x_{ijk}, \text{ and} \tag{2.24}$$

$$x^*_{ijk} = 0 \qquad \text{for } k = 1, ..., max_j - 1. \tag{2.25}$$

Further, for each $j \in B \setminus D$, we set

$$y^*_{j,max_j} = y_{j,max_j} + \frac{\sum_{k=1}^{max_j - 1} \sum_{i \in G} x_{ijk}}{u_{j,max_j}}, \text{ and} \tag{2.26}$$

$$y^*_{jk} = 0 \qquad \text{for } k = 1, ..., max_j - 1. \tag{2.27}$$

All other variables remain the same, that is

$$x^*_{ijk} = x_{ijk} \qquad \forall i \in G, j \in D, k \in Z_j. \tag{2.28}$$

It is obvious that the costs of (x^*, y^*) cannot exceed the costs of (x, y) since the total amount of items has remained the same for each supplier, while in (x^*, y^*) all items are purchased in the highest interval (and we have $c_{i,j,max_j} \leqslant c_{ijk}$ $\forall i, j, k$, see (2.2)). Let us now argue that (x^*, y^*) is a feasible solution of GENTQDF.

Evidently, (x^*, y^*) satisfies (2.13), (2.17), (2.18), (2.19), (2.20) and (2.21). To show that (x^*, y^*) satisfies ((2.14)) and ((2.22)), we need to show that $y^*_{j,max_j} \leqslant 1$ for $j \in B \setminus D$. Observe that for $k = 1, ..., max_j - 1$ we have $\sum_{i \in G} x_{ijk}/u_{jk} \leqslant y_{jk}$ (using the feasibility of (x, y) with respect to ((2.16))) and thus

$$\sum_{i \in G} \frac{x_{ijk}}{u_{i,max_j}} \leqslant y_{jk} \qquad \text{for } k = 1, ..., max_j - 1.$$

Summing over $k = 1, ..., max_j - 1$ implies that $\sum_{k=1}^{max_j - 1} (\sum_{i \in G} x_{ijk})/u_{j,max_j} \leqslant \sum_{k=1}^{max_j - 1} y_{jk}$ and together with the feasibility of (x, y) with respect to (2.14), this leads to (x^*, y^*) satisfying (2.14) and (2.22).

Consider now for some $j \in B \setminus D$ constraints (2.15), written alternatively as $\sum_{i \in G} x_{ijk} \geqslant l_{jk} y_{jk}$ for $k = 1, ..., max_j$. In case $k < max_j$, the right-hand side equals 0 (since $y^*_{jk} = 0$ for $k < max_j$ by construction) and feasibility follows. In case $k = max_j$, we have, using feasibility of (x, y), that

$$\sum_{i \in G} x_{i,j,max_j} \geqslant l_{j,max_j} y_{j,max_j}. \tag{2.29}$$

Also it is true that

$$\sum_{k=1}^{max_j - 1} \sum_{i \in G} x_{ijk} \geqslant \frac{\sum_{k=1}^{max_j - 1} \sum_{i \in G} x_{ijk}}{u_{j,max_j}} l_{j,max_j}. \tag{2.30}$$

Summing (2.29) and (2.30) yields:

$$\begin{aligned} \sum_{i \in G} x^*_{i,j,max_j} &= \sum_{i \in G} (x_{i,j,max_j} + \sum_{k=1}^{max_j - 1} x_{ijk}) \\ &\geqslant l_{j,max_j} (y_{j,max_j} + \sum_{k=1}^{max_j - 1} \sum_{i \in G} \frac{x_{ijk}}{u_{j,max_j}}) = l_{j,max_j} y^*_{j,max_j}. \end{aligned} \tag{2.31}$$

Thus (x^*, y^*) satisfies constraints (2.15).

To verify that (x^*, y^*) satisfies constraints (2.16), observe that for $j \in B \setminus D$ and for $k = 1, ..., max_j - 1$, we have $\sum_{i \in G} x^*_{ijk} = 0$ and $y^*_{jk} = 0$ (this follows by construction of x^* and y^*). Finally, in case $k = max_j$ we have

$$\sum_{i \in G} x_{i,j,max_j} \leqslant u_{j,max_j} y_{j,max_j}, \text{ and} \tag{2.32}$$

$$\sum_{k=1}^{max_j - 1} \sum_{i \in G} x_{ijk} = \frac{\sum_{k=1}^{max_j-1} \sum_{i \in G} x_{ijk}}{u_{j,max_j}} u_{j,max_j}. \tag{2.33}$$

Summing (2.32) and (2.33) yields

$$\begin{aligned} \sum_{i \in G} x^*_{i,j,max_j} &= \sum_{i \in G} (x_{i,j,max_j} + \sum_{k=1}^{max_j-1} x_{ijk}) \\ &\leqslant u_{j,max_j} (y_{j,max_j} + \sum_{k=1}^{max_j-1} \sum_{i \in G} \frac{x_{ijk}}{u_{j,max_j}}) = u_{j,max_j} y^*_{j,max_j}, \end{aligned} \tag{2.34}$$

which shows that constraints (2.16) are also satisfied by (x^*, y^*) and allows us to conclude that (x^*, y^*) is indeed a feasible solution of GENTQDF.

Property 2.23 allows us to simplify formulation (2.12)-(2.22) to the following formulation, in which the y-variables no longer appear.

minimize

$$\sum_{i \in G} \sum_{j \in D} c_{i,j,s(j)} x_{i,j,s(j)} + \sum_{i \in G} \sum_{j \in B \setminus D} c_{i,j,max_j} x_{i,j,max_j} \tag{2.35}$$

subject to

$$\sum_{j \in D} x_{i,j,s(j)} + \sum_{j \in B \setminus D} x_{i,j,max_j} = d_i \qquad \forall i \in G \tag{2.36}$$

$$\sum_{i \in G} x_{i,j,s(j)} - l_{j,s(j)} \geqslant 0 \qquad \forall j \in D \tag{2.37}$$

$$\sum_{i \in G} x_{i,j,s(j)} - u_{j,s(j)} \leqslant 0 \qquad \forall j \in D \tag{2.38}$$

$$\sum_{i \in G} x_{i,j,max_j} - u_{j,max_j} \leqslant 0 \qquad \forall j \in B \setminus D \tag{2.39}$$

$$x_{i,j,max_j} \geqslant 0 \qquad \forall i \in G, j \in B \setminus D \tag{2.40}$$

$$x_{ijk} = 0 \qquad \forall i \in G, j \in B \setminus D, k \neq max_j \tag{2.41}$$

$$x_{ijk} = 0 \qquad \forall i \in G, j \in D, k \neq s(j) \tag{2.42}$$

Let us now build the network. We have three sets of nodes: there is a node for each supplier (a ‘supplier node’), there is a node for each item (an ‘item node’) and there is a single source node. The supply of the source node equals $\sum_{i \in G} d_i$ and the demand of each item node equals d_i. All other demands are 0. Furthermore, there is an arc from the source node to each supplier node. If this supplier is in D, the corresponding lower and upper bounds of this arc are $l_{j,s(j)}$ and $u_{j,s(j)}$; if this supplier is not in D, the lower and upper bounds are 0 and u_{j,max_j}. The choice for a lower bound of 0 for suppliers not in D, even if $l_{j,1}$ is strictly positive, may seem surprising at first sight. It can however be verified that because the y-values are relaxed in GENTQDF, $l_{j,1}$ no longer constrains the x-values. The cost of an arc between the source node and each supplier node equals 0. There are also arcs from each supplier node to each item node. These arcs are not constrained by lower or upper bounds, but do have a cost equal to $c_{i,j,s(j)}$ if the corresponding supplier is in D and equal to c_{i,j,max_j} if this supplier is not in D. This completes the description of the min-cost flow instance. A schematic representation is given in Figure 2.1.

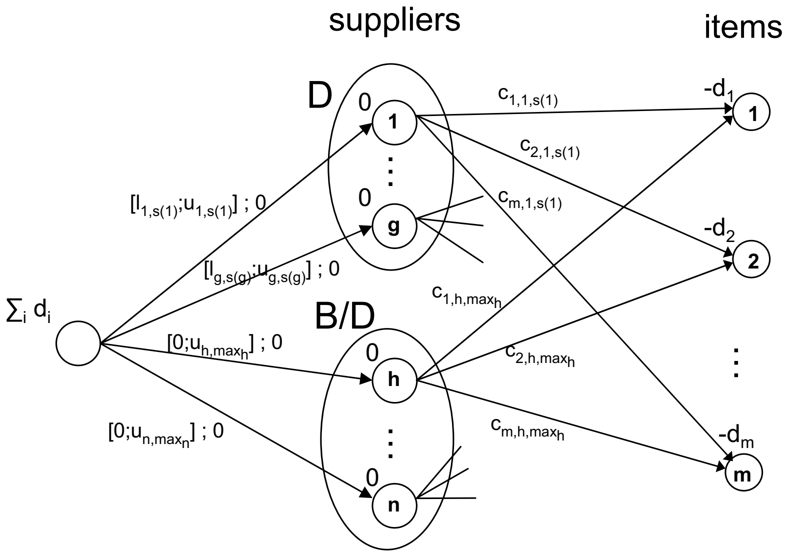

Figure 2.1: GENTQDF as min-cost flow

A solution of this min-cost flow instance is characterized by flows f_{ji} on each arc from supplier j to item i. It corresponds to a solution of GENTQDF as follows:

$$x_{i,j,s(j)} = f_{ji} \qquad \forall i \in G, j \in D, \tag{2.43}$$

$$x_{i,j,max_j} = f_{ji} \qquad \forall i \in G, j \notin D, \tag{2.44}$$

$$y_{j,s(j)} = 1 \qquad \forall j \in D, \tag{2.45}$$

$$y_{j,max_j} = \sum_{i \in G} \frac{f_{ji}}{u_{j,max_j}} \qquad \forall j \notin D. \tag{2.46}$$

All other x- and y-variables of GENTQDF are set equal to 0.

Given (2.23), we conclude that an optimal solution of the min-cost flow problem in Figure 2.1 corresponds to an optimal solution of GENTQDF. It can now easily be seen that an optimal solution of GENTQDF also corresponds to an optimal flow in the min-cost flow problem. Thus, we have shown how GENTQDF can be polynomially transformed to min-cost flow. □

Notice that property (2.23) hints that TQDF is a rather weak formulation. Indeed, when solving the LP-relaxation, only the prices in the bidder's most interesting bid are considered. Moreover, only the upper bound of this bid is respected; constraints (2.7) do not force the solution towards reaching the lower bound in any way. Notice also that as a consequence of Theorem 5, the LP-relaxation of TQDF can be found by solving a min-cost flow problem. This result is the foundation for an exact algorithm to be discussed in chapter 4.

2.6 Conclusion

We presented a multi-item, multi-unit combinatorial auction, tailored for a procurement problem where suppliers adopt a discount that depends on the total quantity ordered. We discussed a number of examples from various domains where suppliers use this discount policy in practice and we compared our auction with other approaches used to solve procurement problems with discounts. We showed that the winner determination problem that results from this auction is NP-hard, and that this is also the case for a number of special cases of this problem. Furthermore, we argued that no polynomial-time algorithm for the TQD problem can achieve a constant worst-case ratio (unless $P = NP$), which contrasts with the case of a single item for which Chauhan et al. (2005) established NP-hardness and gave a fully polynomial time approximation scheme. Finally, we proved that (a generalization of) the linear programming relaxation of a straightforward formulation of the problem can be solved by solving a min-cost flow problem. Thus, we showed that a combinatorial algorithm solves the LP-relaxation of the TQD problem.

Chapter 3

Variants of the TQD auction

When procuring items, other considerations besides the price can be relevant. Although the total quantity discount auction does not incorporate criteria like quality or reliability, we now consider a number of variants of the TQD problem that are common in both procurement practice and literature. A first variant adds constraints on the amount of items the buyer is willing to purchase from a supplier (section 3.1). In another variant (section 3.2), the buyer is allowed to buy more units of any item than strictly needed, while the third variant (section 3.3) imposes a restriction on the number of winning suppliers (suppliers that end up selling some amount of any of the items are called winning suppliers). Finally, a variant that incorporates a multi-period perspective with inventory costs is described (section 3.4). We show that results similar to that of Theorem 5 hold for each of these variants.

3.1 Market share constraints

Suppose that the buyer wants to impose upper and/or lower bounds on the amount of an item that must be ordered from a supplier. Forcing that some supplier j must be allocated an amount of at least q_{ij} and at most Q_{ij} of item i can be done by adding the following constraint to GENTQDF:

$$q_{ij} \leqslant \sum_{k \in Z_j} x_{ijk} \leqslant Q_{ij}. \tag{3.1}$$

On a more global level the buyer could provide bounds on the total allocation for a supplier, across all items. Forcing the total amount of items purchased from a supplier j to lie between w_j and W_j can be done by adding the following constraint to GENTQDF:

$$w_j \leqslant \sum_{i \in G} \sum_{k \in Z_j} x_{ijk} \leqslant W_j. \tag{3.2}$$

These market share constraints are often mentioned in literature (see Davenport & Kalagnanam (2002), Eso et al. (2005), Hohner et al. (2003), Katz et al. (1994)). Notice that none of these extra constraints invalidate property (2.23). Constraints (3.1) can easily be implemented in the min-cost flow graph by changing the lower and upper bounds of the arcs from supplier j to item i. Constraints (3.2) can be realized via the lower and upper bounds of the arcs from the root node to supplier j. Thus, we obtain the following statement:

Theorem 6. *GENTQDF with constraints (3.1) and/or (3.2) can be polynomially transformed to min-cost flow.*

3.2 More-for-less

As described in section 2.5.1, it can be advantageous to obtain more of some item i than the required amount d_i, since this might allow the buyer to use the cheaper prices of a higher interval (see also Crama et al. (2004), Sadrian & Yoon (1994)). If we wish to allow this, constraints (2.13) in GENTQDF should be replaced by

$$\sum_{j \in B} \sum_{k \in Z_j} x_{ijk} \geqslant d_i \qquad \forall i \in G. \tag{3.3}$$

Notice that for the special case where $D = \varnothing$, all units are already bought in the highest intervals in an optimal solution of GENTQDF (see (2.23)). Therefore, there is no need to buy more than d_i of any item i, and an optimal solution can be found by solving the min-cost flow problem in Figure 2.1. In general however, we can formulate the following result:

Theorem 7. *GENTQDF with constraints (2.13) replaced by (3.3) can be polynomially transformed to min-cost flow.*

Proof. Consider the graph in Figure 3.1. It has supplier and item nodes, with demands and connecting arcs like in Figure 2.1. The lower and upper bounds and the costs for these arcs are the same as in Figure 2.1 but in order not to overload the figure, they have been omitted. There is however also a dummy node, corresponding to the additional units of any item i that are bought once the demand d_i is fulfilled. The dummy node has a demand of M, being at least $\sum_{j \in D} l_{j,s(j)}$. The supply of the source node is increased by this same amount M. Furthermore, there is an arc from the source node to the dummy node with cost 0 and an upper bound of M. Notice that any flow in the network in Figure 2.1 is still a feasible flow in the network in Figure 3.1. There are also arcs from each supplier $j \in D$ to the dummy node. These arcs have a cost equal to the price of the supplier's cheapest item in its selected interval $s(j)$. In Figure 3.1, we refer to this item as $q(j)$, i.e. $q(j) = \text{argmin}_i \, c_{i,j,s(j)}$. Notice that this is the item of which we will buy additional units from that supplier to reach the threshold of a higher interval; it would be pointless to buy a more expensive item instead to achieve this. There are no arcs to the dummy node from suppliers not in D. Since for these suppliers the items are already bought at their lowest prices (see (2.23)), there is no use in buying additional items.

Observe that in GENTQDF it can happen that because of the interval selections made for suppliers in D, no feasible solution exists. This is the case if the demands d_i are not high enough to reach the required lower bounds of

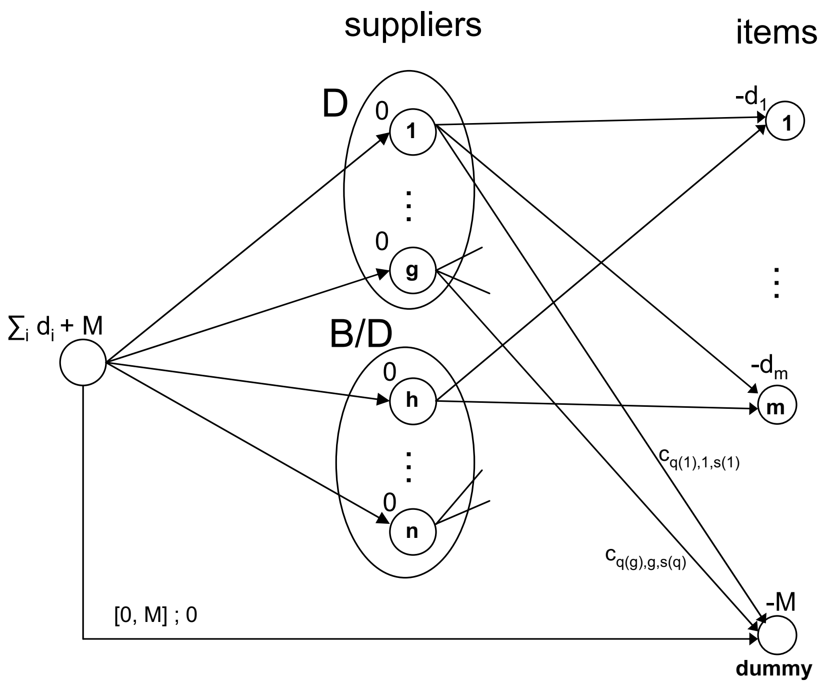

Figure 3.1: GENTQDF with more-for-less as min-cost flow

the selected intervals. In the more-for-less variant of GENTQDF, however, this is no longer possible since it is allowed to buy more than the amounts d_i. Indeed, these extra amounts correspond to the flows on the arcs from suppliers in D to the dummy node. If we refer to the flow from a supplier j to the dummy node as f_{jd}, then a solution of the min-cost flow model in Figure 3.1 corresponds to a solution of GENTQDF with constraints (2.13) replaced by (3.3) as follows:

$$x_{i,j,s(j)} = f_{ji} \qquad \forall i \in G \setminus \{q(j)\}, j \in D, \tag{3.4}$$

$$x_{q(j),j,s(j)} = f_{j,q(j)} + f_{jd} \qquad \forall j \in D, \tag{3.5}$$

$$x_{i,j,max_j} = f_{ji} \qquad \forall i \in G, j \notin D, \tag{3.6}$$

$$y_{j,s(j)} = 1 \qquad \forall j \in D, \tag{3.7}$$

$$y_{j,max_j} = \sum_{i \in G} \frac{f_{ji}}{u_{j,max_j}} \qquad \forall j \notin D. \tag{3.8}$$

□

Until now, we implicitly made the assumption that the buyer can simply buy more than what is demanded and enjoy a higher discount without any further consequence. However, as described in Crama et al. (2004), in practice, overbuying often leads to an extra cost for the buyer. The buyer may for instance need extra storage capacity. Furthermore, the buyer may not be able to use the additional items as profitably as the items of the original demand, or even be forced to pay a cost for the disposal of these items. Let us assume that p_i is this non-negative cost the buyer incurs for each additional unit of item i, in addition to the purchasing cost. Let us define x'_{ijk} as the amount of item i that is bought in addition to the demand in the k-th interval of supplier j. We can now generalize more-for-less GENTQDF as follows:

minimize

$$\sum_{i \in G} \sum_{j \in B} \sum_{k \in Z_j} (c_{ijk} x_{ijk} + (c_{ijk} + p_i) x'_{ijk}) \tag{3.9}$$

subject to

$$\sum_{j \in B} \sum_{k \in Z_j} x_{ijk} = d_i \qquad \forall i \in G \tag{3.10}$$

$$\sum_{k \in Z_j} y_{jk} \leqslant 1 \qquad \forall j \in B \setminus D \tag{3.11}$$

$$\sum_{i \in G} (x_{ijk} + x'_{ijk}) - y_{jk} l_{jk} \geqslant 0 \qquad \forall j \in B \setminus D, k \in Z_j \tag{3.12}$$

$$\sum_{i \in G} (x_{ijk} + x'_{ijk}) - y_{jk} u_{jk} \leqslant 0 \qquad \forall j \in B \setminus D, k \in Z_j \tag{3.13}$$

$$\sum_{i \in G}(x_{i,j,s(j)} + x'_{i,j,s(j)}) - l_{j,s(j)} \geqslant 0 \qquad \forall j \in D \tag{3.14}$$

$$\sum_{i \in G}(x_{i,j,s(j)} + x'_{i,j,s(j)}) - u_{j,s(j)} \leqslant 0 \qquad \forall j \in D \tag{3.15}$$

$$x_{ijk} \geqslant 0 \qquad \forall i \in G, j \in B \setminus D, k \in Z_j \tag{3.16}$$

$$x'_{ijk} \geqslant 0 \qquad \forall i \in G, j \in B \setminus D, k \in Z_j \tag{3.17}$$

$$x_{ijk} = x'_{ijk} = 0 \qquad \forall i \in G, j \in D, k \neq s(j) \tag{3.18}$$

$$0 \leqslant y_{jk} \leqslant 1 \qquad \forall j \in B \setminus D, k \in Z_j \tag{3.19}$$

Consider a min-cost flow network like the one in Figure 3.1, but with the difference that the cost on the arcs from supplier $j \in D$ to the dummy node equals $c_{q(j),j,s(j)} + p_{q(j)}$, with

$$q(j) = \underset{i}{\operatorname{argmin}}(c_{i,j,s(j)} + p_i). \tag{3.20}$$

Let us now argue how a solution of this min-cost flow network corresponds to a solution of generalized more-for-less GENTQDF. It is clear that for suppliers not in D, it remains pointless to buy any additional item, since the buyer can already get the lowest possible price by ordering in the highest intervals. Notice that property (2.23) thus remains valid. For suppliers for which a winning bid has been prespecified, it can be necessary to buy additional items, namely if the demands d_i are insufficiently high to reach the lower bounds of the selected intervals. In this case, the buyer will obviously buy the cheapest additional item, namely the item for which $c_{i,j,s(j)} + p_i$ is minimal. Notice that this is exactly how we defined $q(j)$. It is now easy to see that a solution f of the min-cost flow network corresponds to a solution of generalized more-for-less GENTQDF as follows:

$$x_{i,j,s(j)} = f_{ji} \qquad \forall i \in G, j \in D, \tag{3.21}$$

$$x_{i,j,max_j} = f_{ji} \qquad \forall i \in G, j \notin D, \tag{3.22}$$

$$x'_{q(j),j,s(j)} = f_{jd} \qquad \forall j \in D, \tag{3.23}$$

$$y_{j,s(j)} = 1 \qquad \forall j \in D, \tag{3.24}$$

$$y_{j,max_j} = \sum_{i \in G} \frac{f_{ji}}{u_{j,max_j}} \qquad \forall j \notin D. \tag{3.25}$$

All other x-, x'- and y-variables are set equal to 0. Hence we have proven the following theorem:

Theorem 8. *The generalization of more-for-less GENTQDF can be polynomially transformed to min-cost flow.*

3.3 Limited number of winning suppliers

Another important consideration apart from cost minimization is to make sure that the demand is not procured from too many suppliers (see also Davenport & Kalagnanam (2002), Eso et al. (2005), Hohner et al. (2003), Katz et al. (1994), Sadrian & Yoon (1994)). Otherwise, overhead costs increase due to managing this large amount of suppliers. Limiting the total number of winning suppliers can be done for the order as a whole (section 3.3.1) or per item (section 3.3.2).

3.3.1 Limited total number of winning suppliers

In order to model the requirement that a limited number of suppliers is selected, we need to understand exactly when a supplier receives a positive amount. This happens when $y_{jk} = 1$ for some k, except possibly when $k = 1$, and $l_{j,1} = 0$; the latter situation refers to the case where interval 1, with a lower bound of 0, is selected. Then a supplier might receive nothing, while there is a y-variable with a positive value. To handle this situation, we 'split' each bid with an interval that has a lower bound of 0 and a positive upper bound into two bids: one bid with an interval with a lower bound and an upper bound of 0 (the dummy bid), and one bid with an interval with a lower bound of 1 and an upper bound equal to the original upper bound (interval 1). Notice that by setting this lower bound to 1, we assume that the demands and the lower and upper bounds are or can be scaled to integers. Thus, we have redefined interval 1 by excluding the option of a zero amount of items. Moreover, we let $y_{j,1}$ correspond to this new interval 1. Obviously, selecting a supplier's dummy bid comes down to not selecting this supplier at all, in which case the supplier can simply be removed from the problem. Selecting another bid of a supplier implies that this is a winning supplier. This approach leads to a set D, containing only winning suppliers. In fact, without loss of generality, we can now focus on constraining the winning suppliers not in D, and limit their number to K by adding the following constraint to GENTQDF:

$$\sum_{j \in B \setminus D} \sum_{k \in Z_j} y_{jk} \leqslant K. \tag{3.26}$$

If we assume that the highest volume interval of every supplier in $B \setminus D$ has the same upper bound, we can prove a similar result to that of Theorem 5. We refer to this common upper bound as u_{max}. Given the fact that in most real-life applications suppliers pose no upper bound at all to the amount of items they are willing to sell, this assumption is quite reasonable.

Theorem 9. *If $u_{max_j} = u_{max}$ $\forall j \in B \setminus D$, then GENTQDF with constraint (3.26) added can be polynomially transformed to a min-cost flow problem.*

Proof. First, notice that property (2.23) remains valid in this setting. Indeed, given the x-values, we can find y-values for each supplier $j \in B \setminus D$ and each volume interval $k \in Z_j$ satisfying constraints (2.15) and (2.16) in the following interval:

$$[\frac{\sum_{i\in G} x_{ijk}}{u_{jk}}, \frac{\sum_{i\in G} x_{ijk}}{l_{jk}}]. \tag{3.27}$$

Naturally, in order to fulfill constraints (2.22), the y-values cannot exceed 1. It is easy to verify that shifting items from a supplier's highest interval to one or more lower intervals can never decrease the total y-value of this supplier. Therefore, constraint (3.26) will never force the optimal solution of GENTQDF away from the highest intervals and property (2.23) still holds.

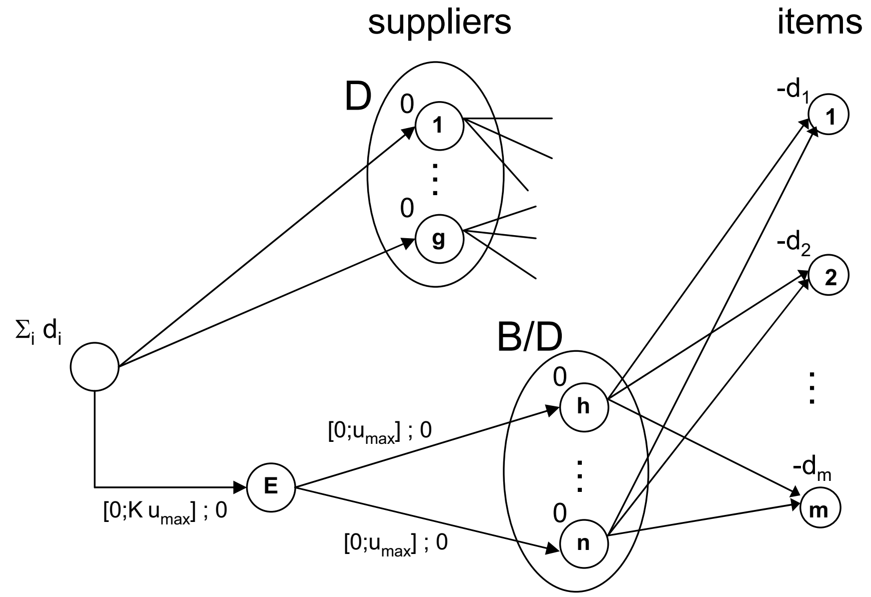

Figure 3.2: GENTQDF with a limited number of winning suppliers as min-cost flow

We can now construct a min-cost flow network (see Figure 3.2). Compared to Figure 2.1, an extra node, referred to as node E, is added. The arc from the root node to node E has an upper bound of Ku_{max}, and the arcs from node E to the supplier nodes have upper bounds of u_{max}.

Let d_{min} be the minimal amount of items that needs to be purchased from suppliers not in D in order to have a feasible solution, i.e., $d_{min} = max(\sum_{i \in G} d_i - \sum_{j \in D} u_{j,s(j)}, 0)$. The min-cost flow problem can only be infeasible if this demand d_{min} is too high for the upper bounds on the arcs, i.e., if $d_{min} > Ku_{max}$. In this case however, GENTQDF with constraint (3.26) is infeasible as well. Indeed, even when choosing the y-values as low as possible, namely as f_j/u_{max}, we fail to meet constraint (3.26):

$$\begin{aligned}\sum_{j \in B \setminus D} \sum_{k \in Z_j} y_{jk} &= \sum_{j \in B \setminus D} f_j/u_{max} \\ &\geqslant d_{min}/u_{max} \\ &> K.\end{aligned}$$

If there exists a feasible flow f to the min-cost flow problem, then we can always find a solution to GENTQDF with constraint (3.26) by setting the x- and y-variables as in (2.43)-(2.46). From Theorem 5, it is clear that this solution satisfies (2.13)-(2.22). Let d_{max} be the maximal amount of items that can be purchased from suppliers not in D in order to keep the solution feasible, i.e., $d_{max} = \sum_{i \in G} d_i - \sum_{j \in D} l_{j,s(j)}$. Obviously, a feasible flow will have $d_{max} \leqslant Ku_{max}$. Therefore, the resulting y-variables will also satisfy (3.26), as shown below:

$$\begin{aligned}\sum_{j \in B \setminus D} \sum_{k \in Z_j} y_{jk} &= \sum_{i \in G} \sum_{j \in B \setminus D} f_{j,i}/u_{max} \\ &\leqslant d_{max}/u_{max} \\ &\leqslant K.\end{aligned}$$

□

Notice that this proof no longer holds when each supplier j has an arbitrary value for u_{max_j}. For instance, if we set the upper bound on the arc from the source to node E equal to the sum of the K highest upper bounds, then it may happen that there exists a feasible flow f such that the corresponding x- and y-variables according to (2.43)-(2.46) are no feasible solution to GENTQDF. Indeed, consider the setting in Figure 3.3, assuming $K = 1$. A flow of 2 to node E, splitting into flows of 1 to supplier 1 and supplier 2 is feasible to the min-cost flow model. However, its corresponding y-values in GENTQDF, 0.5 and 1 respectively, clearly violate constraint (3.26). Analogously, setting the upper bound of the arc to node E equal to the sum of the K lowest upper bounds results in the existence of a solution of GENTQDF for which the corresponding flow is no feasible solution of the min-cost flow model.

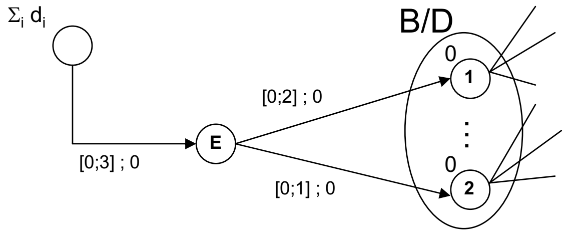

Figure 3.3: Necessity of common u_{max_j}

Property (2.23) is crucial for the possibility to use min-cost flow to solve LP-relaxations of GENTQDF-type formulations. For instance, one could also argue that the number of winning suppliers must be at least a minimum number, say L. Indeed, depending on too few suppliers could move the buyer in a vulnerable position if one of these suppliers is unable to supply as agreed. This could be encoded by adding the following constraint to GENTQDF:

$$\sum_{j \in B} \sum_{k \in Z_j} y_{jk} \geqslant L. \tag{3.28}$$

Property (2.23) is however no longer valid in this setting, since constraint (3.28) pushes the optimal solution away from the highest intervals. Indeed, moving the items towards one or more lower intervals can increase the total y-value of each supplier. This is illustrated by the following example.

	Supplier A	Supplier B	Supplier C	
Interval	1-10	1-10	1-5	6-10
Unit cost	5	1	3	2

Consider a setting where 14 units of one single item need to be bought from three suppliers with volume intervals and costs as indicated in the table above. Also, we wish to order from at least 2 suppliers ($L = 2$). Solving GENTQDF for this example results in the following optimal solution:

$$\begin{aligned} x_A &= 0 & y_A &= 0 \\ x_B &= 10 & y_B &= 1 \\ x_{C1} &= 0.4 & y_{C1} &= 0.4 \\ x_{C2} &= 3.6 & y_{C2} &= 0.6 \end{aligned}$$

It is clear that property (2.23) is not valid for this solution, since it makes use of supplier C's lowest interval. Especially the fact that the optimal solution makes use of more than one interval per supplier, prevents us from following a similar reasoning as in Theorem 5 to transform this variant to a min-cost flow problem.

3.3.2 Limited number of winning suppliers per item

Suppose now that the buyer is interested in limiting the number of winning suppliers for one or more specific items only. Forcing that item i can be supplied by at most Q_i suppliers can be done by adding the following constraints to TQDF:

$$z_{ij} \leqslant \sum_{k \in Z_j} x_{ijk} \leqslant M_{ij} z_{ij} \qquad \forall i \in G, j \in B \qquad (3.29)$$

$$\sum_{j \in B} z_{ij} \leqslant Q_i \qquad \forall i \in G \qquad (3.30)$$

$$z_{ij} \in \{0, 1\} \qquad \forall i \in G, j \in B. \qquad (3.31)$$

We introduced a new binary variable z_{ij} which is 1 if supplier j procures at least 1 unit of item i and 0 otherwise. This is guaranteed by constraints (3.29) and (3.31). In constraint (3.29), the parameter M_{ij} can be set equal to $min(d_i, u_{max_j})$. Constraints (3.30) state that no more than Q_i suppliers should procure item i. We refer to TQDF with constraints (3.29)-(3.31) added as TQDF'.

We now generalize TQDF' by assuming that for some suppliers an interval is prespecified. Additionally, we assume that some z_{ij} variables get value 1 beforehand, some z_{ij} variables get value 0, whilst for others no value is prespecified. When also the y- and z-variables are relaxed, so that they can take any value between 0 and 1, this results in a relaxation of this generalization of TQDF', to which we refer as GENTQDF'.

It can easily be verified that property (2.23) remains valid for GENTQDF'. Indeed, also in this setting we can improve any solution that makes use of intervals other than the highest by shifting items bought in these intervals to the highest interval. As we argued in Theorem 5, it is always possible to adjust the y-variables in such a way that the solution remains feasible. Furthermore, this shift has no influence at all on the z-variables, since the x-variables are summed over all intervals in constraints (3.29).

We can now construct a min-cost flow network like in Figure 2.1. However, for this variant, the arc from a supplier j to an item node i has a lower bound of 1 and an upper bound of M_{ij} if supplier j is chosen to be one of the Q_i suppliers that will be procuring item i. On the other hand, if supplier j is chosen not to be part of the winning suppliers for item i, the arc from node j to node i is deleted. A solution of this min-cost flow problem is characterized by flows f_{ji} on each arc from supplier j to item i. This solution corresponds to the x- and y-variables of the optimal solution of GENTQDF' as indicated in (2.43) to (2.46). The z-variables in GENTQDF' follow from the min-cost flow solution as follows:

$$z_{ij} = \frac{f_{ji}}{M_{ij}} \qquad \forall i \in G, j \in B. \tag{3.32}$$

Indeed, constraints (3.30) force the z-variables towards the lowest value they can get, which is $\sum_{k \in Z_j} x_{ijk}/M_{ij}$. However, from property (2.23), it follows that $\sum_{k \in Z_j} x_{ijk}$ equals $x_{i,j,s(j)}$ for suppliers in D and x_{i,j,max_j} for those not in D, which is exactly f_{ji} (see (2.43) and (2.44)). Thus we obtain the following statement:

Theorem 10. *GENTQDF' can be polynomially transformed to min-cost flow.*

3.4 Multi-period procurement

A lot of research on quantity discount policies has been done in the context of lot sizing problems (see e.g. Xu, Lu & Glover (2000)). Lot sizing problems typically deal with when to order what amount of items and include inventory costs. Whereas in the basic TQD auction we assumed a single-period perspective, we generalize to an auction that handles a multi-period procurement problem in this variant. Indeed, it no longer suffices for the buyer to decide what items to purchase from what supplier, but the buyer also needs to decide when to order what items, taking into account the inventory costs.

We define P as a series of r periods, indexed by p. For each item i, d_{ip} is now the demand for item i in period p. We also define h_{ip} as the cost of holding one unit of item i in inventory at the end of period p. Each bid k is valid only in a period p and consists of a lower bound l_{jkp} and an upper bound u_{jkp}, and values c_{ijkp} as the price of purchasing one unit of item i in period p according to bid k by supplier j. In order to model this variant, we need to generalize the x- and y-variables with an extra index p, referring to the period in which the item is bought. We also generalize the set D to D_p, being the set of suppliers for which an interval has been prespecified for the period p. We refer to this interval as $s(j,p)$. We also introduce the variable v_{ip} as the inventory of item i at the end of period p. The generalized formulation, to which we refer as multi-period GENTQDF then looks as follows:

minimize

$$\sum_{i\in G}\sum_{j\in B}\sum_{k\in Z_j}\sum_{p\in P} c_{ijkp}x_{ijkp} + \sum_{i\in G}\sum_{p\in P} h_{ip}v_{ip} \tag{3.33}$$

subject to

$$v_{i,1} = \sum_{j\in B}\sum_{k\in Z_j} x_{i,j,k,1} - d_{i,1} \qquad \forall i \in G \tag{3.34}$$

$$v_{ip} = v_{i,p-1} + \sum_{j\in B}\sum_{k\in Z_j} x_{ijkp} - d_{ip} \qquad \forall i \in G, p \in P \tag{3.35}$$

$$\sum_{k\in Z_j} y_{jkp} \leqslant 1 \qquad \forall j \in B \setminus D_p, p \in P \tag{3.36}$$

$$\sum_{i\in G} x_{ijkp} - y_{jkp}l_{jkp} \geqslant 0 \qquad \forall j \in B \setminus D_p, k \in Z_j, p \in P \tag{3.37}$$

$$\sum_{i\in G} x_{ijkp} - y_{jkp}u_{jkp} \leqslant 0 \qquad \forall j \in B \setminus D_p, k \in Z_j, p \in P \tag{3.38}$$

$$\sum_{i \in G} x_{i,j,s(j,p),p} - l_{j,s(j,p),p} \geqslant 0 \quad \forall j \in D_p \tag{3.39}$$

$$\sum_{i \in G} x_{i,j,s(j,p),p} - u_{j,s(j,p),p} \leqslant 0 \quad \forall j \in D_p \tag{3.40}$$

$$x_{ijkp} \geqslant 0 \quad \forall i \in G, j \in B \setminus D_p, k \in Z_j, p \in P \tag{3.41}$$

$$x_{ijkp} = 0 \quad \forall i \in G, j \in D_p, k \neq s(j,p), p \in P \tag{3.42}$$

$$0 \leqslant y_{jkp} \leqslant 1 \quad \forall j \in B \setminus D_p, k \in Z_j, p \in P \tag{3.43}$$

Generalizing from (2.23), we claim that there exists an optimal solution (x^*, y^*) of multi-period GENTQDF in which for each $p \in P$ and for each $j \in B \setminus D_p$:

$$\begin{aligned} x^*_{ijkp} &= 0 \quad \forall i \in G, k \neq max_j, \text{ and} \\ y^*_{jkp} &= 0 \quad \forall k \neq max_j. \end{aligned} \tag{3.44}$$

Notice that this claim can be proven in a similar way as (2.23).

Let us now construct a min-cost flow problem similar to the one in Figure 2.1, but now there are supplier nodes (j,p) for each supplier j in each period p. Also, there are item nodes for each item in each period. Each item node (i,p), corresponding with item i in period p, has a demand of d_{ip}. The source node has a supply equal to $\sum_{i \in G} \sum_{p \in P} d_{ip}$. There are arcs from a supplier node (j,q) to an item node (i,r) if $q \leqslant r$. These arcs have a cost equal to $c_{i,j,s(j,q),q} + \sum_{p=q}^{r-1} h_{ip}$ if the corresponding supplier is in D_q and equal to $c_{i,j,max_j,q} + \sum_{p=q}^{r-1} h_{ip}$ if this supplier is not in D_q. A schematic representation is given in Figure 3.4. However, in order not to overload the figure, only one supplier and one item are drawn. Also, we assume that for this supplier no interval is prespecified for any period.

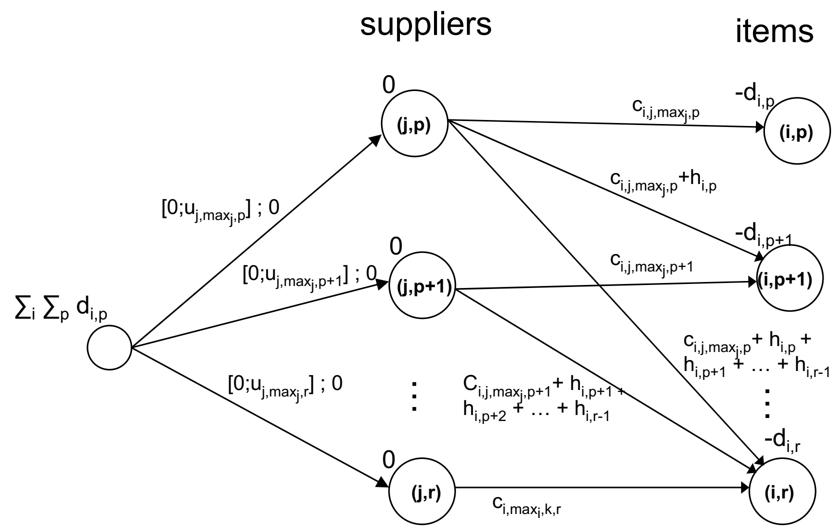
suppliers
items
0
(j,p)
$c_{i,j,max_j,p}$
$-d_{i,p}$
(i,p)
$c_{i,j,max_j,p}+h_{i,p}$
$[0;u_{j,max_j,p}]$; 0
0
$-d_{i,p+1}$
$c_{i,j,max_j,p+1}$
$[0;u_{j,max_j,p+1}]$; 0
(j,p+1)
(i,p+1)
$\Sigma_i \Sigma_p d_{i,p}$
$c_{i,j,max_j,p}+ h_{i,p} + h_{i,p+1} + \ldots + h_{i,r-1}$
$[0;u_{j,max_j,r}]$; 0
$C_{i,j,max_j,p+1}+ h_{i,p+1} + h_{i,p+2} + \ldots + h_{i,r-1}$
$-d_{i,r}$
0
(j,r)
(i,r)
$c_{i,max_j,k,r}$

Figure 3.4: Multi-period GENTQDF as min-cost flow

A solution of the min-cost flow network in Figure 3.4 is characterized by flows f_{jp} from the source node to each supplier node (j,p) and by flows f_{jpiq} from supplier node (j,p) to item node (i,q). This solution can be written as a solution of multi-period GENTQDF as follows:

$$x_{i,j,s(j,p),p} = \sum_{q=p}^{r} f_{jpiq} \qquad \forall i \in G, j \in D, p \in P, \tag{3.45}$$

$$x_{i,j,max_j,p} = \sum_{q=p}^{r} f_{jpiq} \qquad \forall i \in G, j \notin D, p \in P, \tag{3.46}$$

$$y_{j,s(j,p),p} = 1 \qquad \forall j \in D_p, \forall p \in P, \tag{3.47}$$

$$y_{j,max_j,p} = \frac{f_{jp}}{u_{j,max_j,p}} \qquad \forall j \notin D_p, \forall p \in P. \tag{3.48}$$

All other x- and y-variables of multi-period GENTQDF are set equal to 0. The v-variables can now be computed from the x-variables using 3.34 and 3.35.

Theorem 11. *Multi-period GENTQDF can be polynomially transformed to min-cost flow.*

3.5 Conclusion

In this chapter, we recognized that in practice, the price of the items is often not the only criterion upon which the supplier selection decision is based. In chapter 2, we assumed that a preselection of suppliers was made, excluding those suppliers from the auction, who do not satisfy whatever criterion the buyer deems important. However, with this approach, a number of the buyer's concerns cannot be dealt with. A buyer may wish to place bounds on the market share that a bidder can or should obtain. The buyer may also want to purchase more units of any item than originally planned, if this leads to a lower total cost. We also consider a setting where the buyer needs to pay a disposal cost for the extra units bought. Furthermore, the buyer may want to impose a maximum number of winning suppliers (per item or in total) to avoid large overhead costs. Finally, in practice, the buyer is often confronted with a procurement problem that keeps recurring over time. In this setting, the buyer also needs to decide when the items should be purchased, finding a balance between discounts and holding costs. We extended the winner determination problem of the total quantity discount auction to take into account these concerns. We also showed that for each of these variants, property (2.23) still holds, which will turn out to be useful for solving the winner determination problem in the next chapter.

Chapter 4

Exact algorithms for the TQD auction

In this chapter we describe the three exact algorithms used to solve the winner determination problem of the TQD auction and its variants. We discuss an LP based branch-and-bound algorithm (section 4.1), a min-cost flow based branch-and-bound algorithm (section 4.2), and a branch-and-cut algorithm (section 4.3). We use these algorithms to solve a number of generated instances and discuss the resulting computational results in section 4.4.

4.1 LP based branch-and-bound

Branch-and-bound is probably the most widely used technique for solving integer programming problems. To find an optimal integral solution, it typically does a stepwise partitioning of the solution space. The branch-and-bound algorithm can be represented by a tree. In the root node, a relaxation of the original problem is solved. A relaxation is a simplification of the problem, for instance by discarding a number of constraints. If this relaxation does not render an integral solution, the solution space needs to be partitioned in two or more mutually exclusive subsets. Each of these subsets is represented by child node. This partitioning phase is what we call branching.

If a node results in an integral solution, no further partitioning is needed. Either, this solution is better than the best solution found so far (i.e. the incumbent) and thus replaces it as incumbent, or it is not, in which case it can be discarded (pruned). If a fractional solution is found, the node needs to be partitioned further unless we can show that none of its descendants can result in a solution better than the incumbent. This can be done by computing a bound on the best solution that can be reached from this node. The objective function value of the node is an example of such a bound. If a node turns out to be infeasible, it can be pruned as well.

If all generated nodes are considered, the algorithm ends and the incumbent is an optimal solution. Thus, a branch-and-bound algorithm implicitly enumerates all possible solutions. Of course, the less nodes are generated, the less computation time will typically be needed to find the optimal solution. Notice that this is to a great extent determined by the way in which the partitioning is done and by the choice of which node to consider next. Furthermore, a tight bound will allow nodes to be pruned quickly, but may also require a lot of computation time. A thorough discussion of branch-and-bound can be found in Wolsey (1998), Johnson, Nemhauser & Savelsbergh (2000), and Linderoth & Savelsbergh (1999).

We apply the branch-and-bound approach to the TQD problem, by relaxing the integrality constraints (2.10), such that in each node a linear programming problem is solved. We refer to this algorithm as the linear programming based branch-and-bound algorithm and implemented it using Ilog Cplex 8.1. To decide on which variable to branch, Ilog Cplex uses strong branching. Strong branching selects a branching variable after evaluating the actual objective degradations that occur when forcing variables to integer values. A variable y with fractional LP value $\hat{y}$ is tested by temporarily introducing a lower bound $\lceil \hat{y} \rceil$ and subsequently an upper bound $\lfloor \hat{y} \rfloor$, and solving the relaxations. In order to reduce the required computation time, only a restricted subset of the variables with fractional values are considered, and a limited number of simplex iterations are performed to estimate the effect on the objective function. Strong branching and other branching rules are discussed in more detail in Achterberg, Koch & Martin (2005). In order to select the node that is to be analyzed first, Ilog Cplex uses best bound search, which means that the node with the best objective function will be selected (generally near the top of the tree).

In order to solve the variants of the TQD auction, it suffices to add or alter the relevant constraints in the formulation, as discussed in chapter 3. The branch-and-bound algorithm can still be performed on the resulting model. Notice however that for the setting with a limited number of suppliers per item, constraints (3.31) should be relaxed.

4.2 Min-cost flow based branch-and-bound

In this section, we discuss a branch-and-bound algorithm, where in every node of the branching tree a min-cost flow problem needs to be solved. This algorithm is based on Theorem 5, which states that GENTQDF can be polynomially transformed to min-cost flow. In the root node, the LP relaxation of the TQD problem is solved as explained in section 2.5.2. If this results in a fractional solution, we branch by selecting a bidder and creating a branch for every bid by this bidder. In the resulting child nodes, we enforce that this bid is a winning bid. In this way, however, we cannot impose that nothing at all is ordered from this bidder, unless he has a bid with a lower bound of zero. Therefore, for bidders without a bid with a lower bound of zero, we create a dummy bid with a lower and an upper bound of zero, and an index of zero. Selecting this bid then enforces that the corresponding supplier is not to be used in the solution.

In order to select a supplier, we compute for each supplier, the sum of its x-values. We now find the bid for which this sum lies between its lower and upper bound, or, if no such bid exists, the bid for which the sum lies closest to either its upper or lower bound. We refer to the interval of this bid as the LP-interval. We can now compute for each supplier its priority as the number of volume intervals minus the index of the LP-interval. Thus, suppliers that express a lot of bids but receive little in the LP-relaxation, are accorded a high priority. We use this priority to build up the search tree, as we start with the supplier with the highest priority, creating branches from the root node for each of its intervals. In the node from the first branch, we fix the LP-interval of the supplier with the highest priority. In the next branch of that level, we fix the interval directly above this interval; in the following branch and still within this level, we fix the interval directly below it and so on (provided that these intervals exist). In the following level of the branching tree we continue with the supplier with the second highest priority, again branching on its intervals as just explained, and so on (see Figure 4.1). If a supplier has only one interval, with a lower bound of zero, there is no need to create a node in the branching tree for this supplier, since we can fix his interval right away. In this way, every level in the branching tree corresponds to a supplier, and there is a branch for every volume interval of that supplier.

In each node, the LP relaxation given the intervals fixed by the branching decisions, is used as an upper bound of the best solution that can still be reached further down the tree. To traverse the tree, we use a standard depth-first search strategy, such that the supplier with the highest priority and its bid lying closest to the LP solution are explored first. Notice that the tree is completely determined after solving the root node. We experimented with recomputing the priorities in each node and thus building the tree dynamically, but it turned out that this did not reduce the size of the tree enough to compensate for the extra time needed to recompute the priorities. We experimented with other priority settings based on the

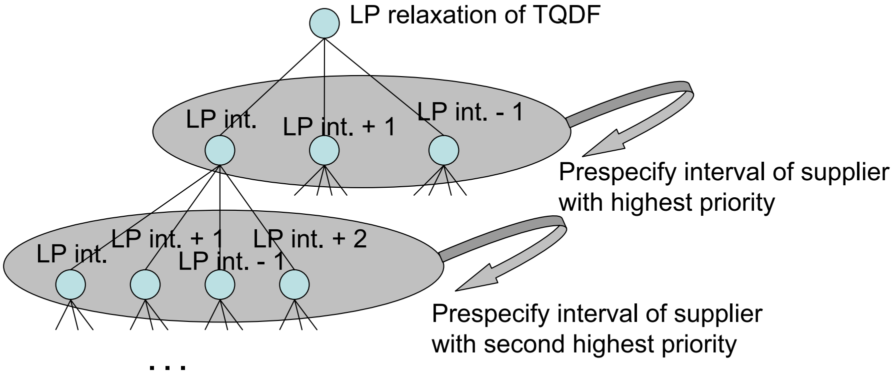
LP relaxation of TQDF
LP int.
LP int. + 1
LP int. - 1
Prespecify interval of supplier with highest priority
LP int.
LP int. + 1
LP int. - 1
LP int. + 2
Prespecify interval of supplier with second highest priority
. . .

Figure 4.1: Branching tree for min-cost flow based branch-and-bound

sum of the x-values and the number of bids, but the choice described above seems to work best. A partial explanation for this observation can be that in the current priority setting, suppliers who receive little are explored first. Given a good solution, the other branches of this supplier should be eliminated by the resulting bound. Finally, we use the solution of the previously solved min-cost flow problem as a starting solution for the next min-cost flow problem that needs to be solved (according to the depth first strategy). Since these problems differ only by some bounds and prices that have been changed, this results in a considerable decrease of the computation time.

The branching tree for both the market share and the more-for-less variant is very similar. In the first variant, we prune the tree by deleting those bids with volume intervals that fall outside the range imposed by the market-share constraints. Afterwards, we can adapt the upper and lower bounds of the highest and lowest interval respectively according to the market share constraints. As a result, the branching tree will typically have less nodes at comparable depths in the market share variant than in the basic case. In the more-for-less variant on the other hand, the branching tree will in general have more nodes at comparable depths compared to its counterpart in the basic case, because less nodes are infeasible in the more-for-less setting. The branching tree for the variant that limits the number of winning suppliers to K differs from the branching tree of the basic case, because for every supplier, we need a dummy interval to impose that this supplier is not to be used in the solution (see also section 3.3). Thus, whereas suppliers with only one interval, which has a lower bound of zero, are left out of the tree completely in the basic case, they now appear in the tree with two branches, representing the decision to buy from that supplier or not. On the other hand, a node needs no further branching as soon as K suppliers have been selected.

The min-cost flow based branch-and-bound algorithm has been programmed in C and compiled using Microsoft Visual C++ 6.0. To solve the min-cost flow problems, we have used the network solver of Ilog Cplex 8.1.

4.3 Branch-and-cut

Branch-and-cut is in fact a combination of branch-and-bound and a cutting plane algorithm. When confronted with a fractional solution, a cutting plane algorithm tries to find a valid inequality that is violated by this fractional solution. The problem of finding such a violated valid equality (if it exists) is known as the separation problem. By repeatedly adding violated constraints, the algorithm converges towards an optimal integral solution. Cutting plane algorithms were first proposed by Gomory (1963); they tend to converge slowly. Moreover, without a special structure, the separation problem may be hard to deal with. For a further discussion of the separation problem, we refer to Wolsey (1998) or Nemhauser & Wolsey (1988).

A cutting plane method can be very useful when combined with a branch-and-bound algorithm. Indeed, cutting planes can be used to strengthen the linear programming relaxations used for bounding. This makes it a much more powerful method than branch-and-bound alone. Whereas cutting planes can result in a considerable reduction in the size of the tree, finding adequate cutting planes can also be very time consuming. This explains why many branch-and-cut algorithms do not perform a cut generation phase in every node of the branching tree. A more thorough discussion of branch-and-cut can be found in e.g. Mitchell (2002).

We implemented the branch-and-cut algorithm by simply using the default settings of the MIP solver of Ilog Cplex 8.1. These default setting include the use of so-called flow cover cuts (see Nemhauser & Wolsey (1988)) that are valid for the TQD problem and its variants.

4.4 Computational results

In this section we discuss the choices that were made to construct the instances on which the algorithms have been tested. We continue with computational results for the TQD problem and its variants and evaluate the performance of our algorithms.

4.4.1 Structure of the instances

In order to test the performance of the exact algorithms, two types of instances have been generated: completely random instances and instances with a special structure, inspired by the instances studied by van de Klundert et al. (2005). All instances have 10, 20, or 50 suppliers and 40 or 100 items. Furthermore, each supplier has a maximum of 3 or 5 bids. For all instances, the total demand for an item is a random number between 1000 and 10000 units. For instances with 40 items, the upper bound increase from one interval to the next is a random number between 10000 and 50000, while for instances with 100 items, the upper bound increase is a random number between 10000 and 100000.

For structured instances, we first determine a base price for each item, randomly picked between 3 and 7. The price for an item in a supplier's first interval is then computed by adding a random number in the interval $[-2, 2]$ to the base price. Furthermore, for each supplier j there is a discount rate $\delta_{jk} \in [0, 0.1]$ for every interval $k > 1$, which determines the price c_{ijk} of item i in interval k as a function of the price in interval $k - 1$ as follows:

$$c_{ijk} = (1 - \delta_{jk})c_{i,j,k-1} \qquad \forall i \in G, j \in B \text{ and } \forall k > 1 \tag{4.1}$$

For random instances, the cost of purchasing an item from a supplier in its first interval is a random number between 2 and 8. The price for this item in each of the next intervals is computed by discounting the price in the previous interval by a percentage picked randomly between 0 and 75%.

The key difference between the random and the structured instances is that for the former instances prices can drop drastically from one interval to the next, whereas for the latter this decrease in price is limited to 10%. Furthermore, for the structured instances, an item that is expensive at one supplier will very likely be expensive at the other suppliers too. For the random instances however, this is not necessarily the case as prices for an item can differ in a wider range between the various suppliers. Finally, the discount percentage one receives when moving from one interval to the next can differ substantially between the items for the random instances, while it is the same for all the items for the structured instances.

In the variant with the market share constraints, only global constraints (as in (3.2)) are included. For the instances with 10 suppliers, 5 suppliers are picked randomly and between 5 and 20 percent of the total demand needs to be purchased from each of those suppliers. For instances with 20 suppliers, we pick 10 suppliers and force between 5 and 15 % of the total demand to go to each of them and for the instances with 50 suppliers this becomes

15 suppliers with each 5 to 10 % of the total demand. The more-for-less variant needs no extra modifications, apart from allowing to buy more than what is demanded. For the third variant, the number of winning suppliers is limited to 5 for all instances. If an instance has no solution with only 5 winning suppliers, the interval thresholds are doubled for each supplier until a solution exists.

4.4.2 Results

The results of our experiments are summarized in tables 4.1 to 4.4. The instances are coded with 'S' for structured and 'R' for random instances. The first number indicates the number of bidders, the second number reflects the number of items and the third number is the maximal number of bids per supplier. For each of these types of instances, 10 instances were generated and solved with the three algorithms. This resulted in computation times (in seconds) and a number of nodes searched in the branching tree for each algorithm, averaged per type of instance in the tables. All computations were done on a Pentium *IV* 2 GHz computer, with 512 Mb RAM.

In Table 4.1, the results for the winner determination problem of the basic TQD auction are presented. Each algorithm solves all instances in a reasonable amount of time; random instances seem to be harder to solve than the structured ones. The min-cost flow based algorithm clearly performs best in terms of computation time for all instances with 10 or 20 suppliers. However, instances with 50 suppliers prove to be harder to solve with this algorithm. The solution time per node is undoubtedly the smallest with the min-cost flow approach (about 10 times smaller than with the linear programming based branch-and-bound algorithm and 100 to up to 2000 times smaller than with the branch-and-cut approach). However, the min-cost flow approach also needs to investigate more nodes than the other two exact algorithms. Comparing the LP based approach with the min-cost flow based approach learns that the former is faster if it needs to explore at least 10 times less nodes. The branch-and-cut approach clearly searches the least amount of nodes, but to achieve this it needs a time-consuming cut generation process. The results show that it pays to generate cuts when the number of suppliers is large.

The results of our experiments with the variant with market share constraints are summarized in Table 4.2. As in the basic case, the random instances require more computation time than the structured ones. Market share constraints are problematic for the branch-and-cut algorithm, whose computation times sometimes even double compared to the basic case. The linear programming based branch-and-bound algorithm deals with these constraints much better, since it manages to solve the instances faster than in the basic case. The results show, however, that the min-cost flow algorithm is the obvious algorithm to deal with instances of this variant. Especially for the instances with 50 suppliers, adding market share constraints causes the computations times to slump compared to the basic case. Moreover, less nodes need to be searched, which can be explained by the construction of the branching tree as described in section 4.2.

Table 4.3 shows the results for the more-for-less variant. It turns out that in none of the structured instances purchasing extra items leads to a lower total cost. In the random instances however, it is profitable in more than 85% of the instances to buy more than strictly needed. This is explained by the fact that discounts are substantially larger for the random cases than for the structured instances (see section 4.4.1). As in the basic case, it seems to be the case that the LP based algorithm is outperformed by the min-cost flow based approach, as long as the latter does not require over 10 times more nodes than the former. Thus, the min-cost flow based algorithm performs best on all instances with 10 or 20 suppliers, but for the instances with 50 suppliers, it is advisable to use the linear programming based branch-and-bound algorithm. Compared to the basic case, the min-cost flow algorithm needs to explore slightly more nodes, resulting in a higher computation time. Apart from the random instances with 50 suppliers, which seem very difficult for all algorithms, this increase in computation time remains very modest. The linear programming based branch-and-bound algorithm is not affected too much either. The branch-and-cut algorithm however deals poorly with this variant.

Finally, Table 4.4 describes the results for the variant that limits the number of winning suppliers. This constraint proved to be binding for more than 98% of the structured instances but less than 50% of the random instances. For the random instances, the prices drop sharper from one interval to the next, which makes it more interesting to go for the higher intervals. This leads to an optimal solution with less suppliers than for the structured instances. This explains why a constraint limiting the number of winning suppliers less often affects the random instances. As for the computation times, branch-and-cut seems the best option for the structured instances. For the random instances, the picture is less clear. The instances with 10 suppliers are best solved with the min-cost flow algorithm, although this algorithm is left far behind by the other two for the instances with 20 suppliers. For these instances, branch-and-bound based on linear programming outperforms the other algorithms for instances where suppliers can have up to 5 volume intervals. Branch-and-cut is the fastest approach to solve random instances with 20 suppliers and up to 3 volume intervals per supplier. Notice that no instances with 50 suppliers are mentioned in this table, because the computation times for these problems were impractically high for all algorithms.

4.5 Conclusion

We described three exact algorithms for solving the winner determination problem of the total quantity discount auction and its variants. One algorithm is based on our result that the problem can be solved by solving a number of min-cost flow problems; the other two algorithms are a branch-and-cut and an linear programming based branch-and-bound algorithm. The algorithms were tested on fairly large randomly generated instances of the basic problem and three variants.

Our computational results show that all three algorithms came to an exact solution in a reasonable amount of time. However, it also became clear that each algorithm has instances for which it performs best. In general, the min-cost flow based algorithm works best for instances where the number of suppliers does not exceed 20 (which seems to correspond to most practical cases). It works especially well for the variant where we imposed constraints on the market share a supplier is allowed to obtain. The branch-and-cut algorithm outperforms the other algorithms on large instances in terms of suppliers of the basic case and on the structured instances of the variant that requires a limited amount of winning suppliers. Finally, the linear programming based branch-and-bound algorithm is at its best with the large instances of the variant where the buyer is allowed to purchase more than strictly needed.

	lp b&b		**mcf b&b**		**branch&cut**	
Instances	time [s]	#nodes	time [s]	#nodes	time [s]	#nodes
S-10-40-3	0.08	29.3	0.01	116.6	0.09	0.3
S-10-40-5	0.11	52.5	0.02	161.1	0.15	10.9
S-10-100-3	0.11	17.2	0.02	69.8	0.12	0.2
S-10-100-5	0.36	74.3	0.14	501.6	0.55	3.2
S-20-40-3	0.16	73.5	0.07	389.2	0.12	0.5
S-20-40-5	0.58	207.3	0.38	1,887.8	0.50	4.7
S-20-100-3	0.57	128.8	0.30	749.6	0.34	1.3
S-20-100-5	1.07	155.1	0.67	1,512.8	1.17	2.1
S-50-40-3	1.92	719.0	5.61	16,671.8	0.51	2.7
S-50-40-5	7.81	2,087.2	32.93	85,210.4	2.99	16.5
S-50-100-3	4.67	696.1	21.81	26,595.3	1.45	2.1
S-50-100-5	24.41	2,614.0	159.77	168,181.3	10.45	14.7
R-10-40-3	0.07	24.3	0.01	54.9	0.09	2.1
R-10-40-5	0.31	160.7	0.07	428.9	0.59	30.5
R-10-100-3	0.10	10.1	0.02	46.9	0.14	2.6
R-10-100-5	0.78	160.0	0.31	845.1	1.50	31.7
R-20-40-3	0.25	121.6	0.14	700.5	0.29	9.5
R-20-40-5	1.56	659.9	0.45	2,155.6	1.81	68.5
R-20-100-3	1.05	235.0	0.59	1,249.6	0.83	8.1
R-20-100-5	5.34	882.8	3.18	3,938.2	6.81	70.1
R-50-40-3	6.17	2,411.0	10.31	28,975.3	1.67	81.6
R-50-40-5	18.41	4,303.1	18.60	48,876.6	14.18	140.9
R-50-100-3	38.47	6,289.1	97.29	103,885.7	8.84	43.8
R-50-100-5	122.49	11,451.2	241.39	237,953.3	61.71	216.8

Table 4.1: Computational results for the basic case

	lp b&b		**mcf b&b**		**branch&cut**	
Instances	time [s]	#nodes	time [s]	#nodes	time [s]	#nodes
S-10-40-3	0.08	25.3	0.01	55.6	0.12	0.7
S-10-40-5	0.12	34.4	0.01	51.3	0.21	2.7
S-10-100-3	0.14	17.6	0.03	102.0	0.18	0.9
S-10-100-5	0.39	58.5	0.07	223.7	0.82	2.9
S-20-40-3	0.20	56.4	0.05	233.6	0.29	0.8
S-20-40-5	0.91	255.8	0.10	453.8	1.68	17.4
S-20-100-3	0.78	136.6	0.13	267.1	0.86	2.5
S-20-100-5	2.08	268.8	0.35	672.4	3.47	9.3
S-50-40-3	1.07	196.1	0.24	622.5	1.18	5.7
S-50-40-5	5.47	790.7	0.31	858.5	9.36	71.9
S-50-100-3	3.58	290.4	1.21	1,185.0	2.86	5.8
S-50-100-5	15.00	807.3	2.85	3,002.7	20.08	63.7
R-10-40-3	0.08	24.0	0.01	67.9	0.15	7.5
R-10-40-5	0.30	125.1	0.04	248.5	0.80	20.4
R-10-100-3	0.14	15.1	0.02	40.7	0.20	0.2
R-10-100-5	1.04	213.2	0.20	546.7	2.46	27.0
R-20-40-3	0.32	132.7	0.12	484.5	0.60	21.6
R-20-40-5	1.50	505.7	0.24	1,062.4	2.91	62.2
R-20-100-3	1.27	253.6	0.26	453.0	1.81	26.5
R-20-100-5	8.44	1,226.5	5.50	9,671.1	11.95	105.2
R-50-40-3	1.15	214.0	0.19	526.5	2.38	25.8
R-50-40-5	7.32	2,099.7	0.56	1,552.2	19.18	273.7
R-50-100-3	3.69	287.5	2.12	2,046.3	7.66	34.1
R-50-100-5	27.79	1,731.3	15.55	15,900.1	59.75	228.9

Table 4.2: Computational results for variant 1 (market share constraints)

	lp b&b		**mcf b&b**		**branch&cut**	
Instances	time [s]	#nodes	time [s]	#nodes	time [s]	#nodes
S-10-40-3	0.06	27.9	0.02	118.7	0.14	0.4
S-10-40-5	0.11	50.6	0.02	166.8	0.49	23.2
S-10-100-3	0.12	17.5	0.02	69.8	0.29	6.8
S-10-100-5	0.40	75.3	0.16	513.3	2.48	66.5
S-20-40-3	0.17	71.3	0.07	389.9	0.45	5.2
S-20-40-5	0.61	219.3	0.43	1,997.3	4.42	148.7
S-20-100-3	0.63	135.7	0.32	749.9	2.08	43.4
S-20-100-5	1.26	185.8	0.74	1,545.9	11.47	164.2
S-50-40-3	1.98	739.7	5.57	16,724.3	8.90	261.9
S-50-40-5	9.88	2,491.3	36.90	94,865.4	147.56	2,500.0
S-50-100-3	5.65	832.3	22.39	26,625.3	28.42	523.7
S-50-100-5	32.31	3,365.0	171.14	172,466.7	271.80	3,006.0
R-10-40-3	0.06	21.6	0.02	55.5	0.10	0.1
R-10-40-5	0.17	82.9	0.05	428.5	0.67	18.5
R-10-100-3	0.10	9.3	0.01	45.6	0.17	0.2
R-10-100-5	0.50	91.5	0.24	1,068.9	2.49	42.6
R-20-40-3	0.24	117.0	0.13	793.8	0.60	15.9
R-20-40-5	0.91	343.9	0.47	3,167.6	3.55	89.8
R-20-100-3	1.00	241.4	0.53	1,369.0	1.85	17.6
R-20-100-5	4.46	801.6	3.11	10,389.8	25.09	434.8
R-50-40-3	10.51	4,615.8	15.91	59,066.0	35.16	1,195.8
R-50-40-5	25.54	6,714.3	39.33	171,566.6	169.82	1,511.1
R-50-100-3	79.93	14,059.4	130.42	206,668.9	274.82	6,035.4
R-50-100-5	398.31	45,945.3	446.85	798,002.9	2,036.07	17,577.4

Table 4.3: Computational results for variant 2 (more for less)

	lp b&b		**mcf b&b**		**branch&cut**	
Instances	time [s]	#nodes	time [s]	#nodes	time [s]	#nodes
S-10-40-3	0.29	372.0	0.26	1,786.3	0.17	3.4
S-10-40-5	0.41	421.1	0.32	2,135.7	0.27	8.1
S-10-100-3	1.23	603.4	0.73	2,017.0	0.35	8.0
S-10-100-5	2.65	1,007.4	1.43	4,115.9	1.17	16.5
S-20-40-3	3.00	2,763.0	6.70	30,788.1	0.34	2.4
S-20-40-5	2.39	1,345.2	15.48	66,377.0	1.07	27.3
S-20-100-3	16.76	5,083.6	23.83	44,780.9	1.63	17.3
S-20-100-5	7.15	1,597.2	20.02	36,588.2	2.73	12.8
R-10-40-3	0.06	25.6	0.03	229.6	0.09	2.1
R-10-40-5	0.26	181.9	0.43	2,697.7	0.57	35.9
R-10-100-3	0.13	26.6	0.10	302.7	0.11	0.0
R-10-100-5	0.74	195.9	0.71	2,040.6	1.33	28.5
R-20-40-3	0.37	252.7	2.06	9,812.6	0.30	8.5
R-20-40-5	1.08	578.6	5.73	25,467.2	1.63	61.2
R-20-100-3	2.38	730.9	7.07	13,881.7	0.91	9.5
R-20-100-5	4.19	920.6	26.47	48,444.9	6.49	66.6

Table 4.4: Computational results for variant 3 (limited nr. of winning suppliers)

Chapter 5

Exclusionary side constraints

In the total quantity discount auction (see chapters 2 to 4), the buyer's task is to select exactly one volume interval for each supplier from whom he wishes to purchase. The choice for a specific volume interval therefore excludes the possibility to purchase items at the prices of any other volume interval this supplier may have. In this chapter, we take a closer look at these so-called *exclusionary side constraints* and we investigate how they contribute to the complexity of the TQD auction. In order to get a clearer view on these constraints, we study them in the context of the transportation problem, which is a well-known and efficiently solvable problem.

5.1 The transportation problem with exclusionary side constraints

The ordinary transportation problem can be described as follows: given a number of supply nodes each with a certain supply of items, a number of demand nodes each with a certain demand for items, and a unit transportation cost for each pair consisting of a supply node and a demand node, send the items from the supply nodes to the demand nodes at a minimum cost. In this chapter we consider the variant where for each demand node a set of pairs of supply nodes is given such that at most one supply node of each given pair is allowed to send items to that demand node. Following the literature, we refer to this problem as the transportation problem with exclusionary side constraints (TPESC).

The transportation problem with exclusionary side constraints can be formulated as follows. Let there be a set S of *supply* nodes, each with a supply of s_i, $i \in S$, and a set D of *demand* nodes, each with a demand of d_j, $j \in D$. For each pair consisting of supply node $i \in S$ and demand node $j \in D$, a unit cost $c_{ij} \geqslant 0$ is given. Finally, for each demand node $j \in D$, a (possibly empty) set of pairs of supply nodes, called F_j, is given; thus $F_j = \{(i_1, i_2)|\ (i_1, i_2) \in S \times S, i_1 \neq i_2\}$. We assume that all data are integral. The problem is to send all supply to the demand nodes at minimum cost, such that each demand node $j \in D$ receives items from at most one supply node for each pair of supply nodes present in F_j. Obviously, if $F_j = \emptyset$ for all $j \in D$, the ordinary transportation problem arises. Notice that we assume that total supply equals total demand, that is $\sum_{i \in S} s_i = \sum_{j \in D} d_j$. For a mathematical formulation, we refer to Sun (2002). When we use the phrase “the feasibility version of TPESC”, we refer to the situation where (i) the (given) bipartite network between supply nodes and demand nodes is not necessarily complete, (ii) no costs are specified, and (iii), the question to answer is whether a feasible solution (using only edges from the network) exists.

As far as we are aware, this problem has first been introduced by Cao (1992), who described an application in storage management of containers. In this application, arriving containers must be positioned in rows of a storage yard, such that the costs of operations (searching, loading, retrieving) are minimized. Differences in size, ownership, or content may disallow containers to be stored in the same row, giving rise to exclusionary side constraints. A branch-and-bound approach was described to solve the problem. Other branch-and-bound approaches are described and tested in Sun (2002), while evolutionary algorithms have been proposed and tested by Cao & Uebe

(1995), and Syarif & Gen (2003). These contributions suggest that those authors believe that the problem is NP-hard, although no formal statement of this result seems to have been made.

In this chapter, we confirm this thesis by showing that the feasibility version of TPESC is NP-complete. More specifically, we establish for each of three relevant special cases of TPESC its complexity status. One special case concerns the setting where all $F_j, j \in D$ are equal and is related to the winner determination problem of the TQD auction, discussed in chapters 2 to 4. We refer to this case as TPESC with *identical* exclusionary sets (see Section 5.2). In section 5.3, we investigate TPESC with a *single* exclusionary set, i.e. a setting where all sets F_j except one are empty. Finally, in Section 5.4 we discuss another special case, where the number of suppliers is fixed.

5.2 TPESC with identical exclusionary sets

In this section we focus on the TPESC with identical exclusionary sets. This special case arises in the total quantity discount auction (see chapters 2 to 4). For the reader's convenience, we now briefly recapitulate the TQD auction and its winner determination problem. Consider a buyer procuring given amounts of different items from different suppliers by means of a combinatorial auction. Suppliers can participate in this auction by submitting at least one bid, consisting of prices charged for each individual item, and a volume interval on the total number of sold items, within which these price are valid. Each of the suppliers thus uses a so-called total quantity discount policy to set the prices for the different items. The resulting winner determination problem is to decide which bids to accept, in order to acquire the given amounts of each of the different items at minimum total cost. Thus, a solution for this problem prescribes how many units of each item are ordered from each supplier.

Observe that the winner determination problem of the TQD auction (as the TPESC problem) is a generalization of the ordinary transportation problem. Indeed, by associating a demand node with each item (with its demand equal to the amount that needs to be bought), and by associating a supply node with appropriate lower and upper bounds with each volume interval of each supplier, the TQD winner determination problem boils down to selecting supply nodes (at most one from each supplier) and finding the right amount of units of each item to be transported. In case a supplier can only deliver a fixed number of items, that is, there is only one supply node for each supplier with coinciding upper and lower bound, the ordinary transportation problem arises. One important aspect in this generalization of the transportation problem is the fact that for each demand node, a set of supply nodes is given (namely the nodes corresponding to the intervals of a single supplier) from which at most one can be used to actually supply that demand node; this corresponds to our F_j sets, with $j \in D$. Observe that if a supplier uses more than two intervals, this is easily accommodated by having an element in the exclusionary set for each pair of volume intervals (which gives rise to a polynomial number of elements in the exclusionary set). Also, observe that these sets are the same for all demand nodes, in other words, we are dealing with an instance of TPESC with identical F-sets.

Let us now formulate our results for this special case of TPESC. We first prove that the problem with $|D| = 2$, that is, the case of two demand nodes, is weakly NP-hard, then we exhibit a pseudo-polynomial time algorithm for this case, and finally we show that the feasibility version of the problem with $|D| = 3$ is strongly NP-complete.

Theorem 12. *The feasibility version of TPESC with identical exclusionary sets is NP-complete, even if $|D| = 2$.*

Proof. We prove the theorem by presenting a reduction from Even-Odd

Partitioning (EOP) to TPESC. EOP is proved to be NP-complete in Garey, Tarjan & Wilfong (1988).

EOP
Input: n pairs of positive integers (x_{2i-1}, x_{2i}), $i = 1, \ldots, n$.
Question: does there exist a partition of $\{1, \ldots, 2n\}$ into disjoint subsets A and B with $|A \cap \{2i-1, 2i\}| = |B \cap \{2i-1, 2i\}| = 1$ for $i = 1, \ldots, n$, and with $\sum_{i \in A} x_i = \sum_{i \in B} x_i$?

For each integer in the input of EOP, we construct a supply node with supply equal to the value of the integer, that is, we set $S = \{1, 2, \ldots, 2n\}$ with $s_i = x_i$ for $i = 1, \ldots, 2n$. There are two demand nodes, each having demand $d_1 = d_2 = \frac{1}{2} \sum_{i=1}^{2n} x_i$. We set $F_1 = F_2 = \{(2i-1, 2i)|\ i = 1, \ldots, n\}$, implying that at most one supply node per pair is allowed to send items to that demand node. Each supply node is connected to each demand node. This completes the description of the instance of TPESC.

A yes-answer to the EOP instance directly corresponds to a feasible solution of the TPESC instance. Also, by observing the fact that the two demand nodes have identical exclusionary constraints, it is clear that in any feasible solution of the TPESC instance, each supply node sends its entire supply to precisely one of the demand nodes which in turn corresponds to a yes-answer of the EOP instance. □

Of course, this result does not rule out the existence of a pseudo-polynomial time algorithm for TPESC with common exclusionary sets and two demand nodes. We will now describe such an algorithm.

We first construct a graph $G = (V, E)$. There is a node in G for each supply node in the TPESC instance. Let there be an edge between each pair of nodes for which there is an exclusionary constraint in F. The resulting graph can be partitioned into a number of connected components (V_i, E_i),

$i = 1, \ldots, c$, such that there is no exclusionary constraint between two vertices in different sets V_i. Thus, any two vertices that are connected by an edge are part of the same component. For each component i, there are only two possible ways of dividing the supply of the nodes of that component over the two demand nodes (as observed earlier, the fact that the two demand nodes have identical exclusionary constraints and that total supply equals total demand, implies that in any feasible solution, each supply node sends its entire supply to precisely one of the demand nodes). Indeed, if we pick an arbitrary node of each component and call it the *primal node* of the component, we can either assign its supply to the first demand node or to the second. This choice determines to which demand nodes the supply of the other nodes of that component must be sent. Notice that we assume here that there exists a feasible way of sending the supply of the supply nodes of a component to the demand nodes, that is, we assume there is no contradiction caused by the exclusionary sets. In a preprocessing phase, we can find out whether a feasible way exists by verifying 2-coloredness of each component.

Suppose that assigning the supply of the primal node of component i to the first demand node results in a total supply of w_i being sent to the first demand node by the component i, and a remaining supply of $r_i = \sum_{j \in V_i} s_j - w_i$ being sent to the second demand node. Further, let p_{i1} (p_{i2}) correspond to the total cost corresponding to component i, when the supply of the primal node is being sent to the first (second) demand node. We partition the set of components into two subsets as follows: $C_1 = \{i | w_i \geqslant r_i\}$, and $C_2 = \{i | w_i < r_i\}$. This allows us to define decision variables x_i such that, for $i \in C_1$: $x_i = 1(0)$ if the supply of the primal node is being sent to the first (second) demand node, and for $i \in C_2$: $x_i = 1(0)$ if the supply of the primal node is being sent to the second (first) demand node. We can formulate the TPESC problem with identical exclusionary sets and two demand nodes as follows:

minimize

$$\sum_{i\in C_1}[p_{i1}x_i + p_{i2}(1-x_i)] + \sum_{i\in C_2}[p_{i2}x_i + p_{i1}(1-x_i)] \tag{5.1}$$

subject to

$$\sum_{i\in C_1}[w_i x_i + r_i(1-x_i)] + \sum_{i\in C_2}[r_i x_i + w_i(1-x_i)] = d_1 \tag{5.2}$$

$$\sum_{i\in C_1}[r_i x_i + w_i(1-x_i)] + \sum_{i\in C_2}[w_i x_i + r_i(1-x_i)] = d_2 \tag{5.3}$$

$$x_i \in \{0,1\} \text{ for } i = 1,\ldots,c. \tag{5.4}$$

Let us now define for $i \in C_1$: $a_i = w_i - r_i$, $p_i = p_{i1} - p_{i2}$, and for $i \in C_2$: $a_i = r_i - w_i$, $p_i = p_{i2} - p_{i1}$, and we define $U = d_1 - \sum_{i\in C_1} r_i - \sum_{i\in C_2} w_i$.

Rewriting (5.1)-(5.4) using these definitions gives us the following equivalent integer program:

minimize

$$\sum_{i=1}^{c} p_i x_i \tag{5.5}$$

subject to

$$\sum_{i=1}^{c} a_i x_i = U \tag{5.6}$$

$$x_i \in \{0,1\} \text{ for } i = 1,\ldots,c. \tag{5.7}$$

Notice that when constraint (5.6) is satisfied, both the first and the second demand node receive their required supply of d_1 respectively d_2. Also observe that the definitions above imply that $a_i \geqslant 0$. In fact, we can eliminate those variables x_i which have as coefficient $a_i = 0$ (since, in an optimal solution we set, in case $a_i = 0$: $x_i = 1$ if $p_i \leqslant 0$, else we set $x_i = 0$). Thus, henceforth we will assume that $a_i \geqslant 1$. Furthermore, we assume that $U \geqslant 0$, since no solution exists if $U < 0$.

This problem is a generalization of the change making problem (see Martello & Toth (1990)), since there is a cost p_i associated to each variable x_i. Furthermore, there are bounds equal to 1 on the variables. Wright (1975) developed a dynamic program for the change-making problem. The following modified version of this algorithm, to which we refer as algorithm DP, provides an optimal solution for formulation (5.5)-(5.7).

Let $f_q(z)$ be the optimal solution value of a sub-instance of (5.5)-(5.7), consisting of components $1, \ldots, q$ and a right-hand side of z, with $1 \leqslant q \leqslant c$ and $0 \leqslant z \leqslant U$. If no solution exists for a combination of values q and z, then $f_q(z) = \infty$. It is clear that

$$f_1(z) = \begin{cases} 0 & \text{if } z = 0; \\ p_1 & \text{if } z = a_1; \\ \infty & \text{if } z \neq a_1. \end{cases}$$

Now, $f_q(z)$ can be computed by considering increasing values of q from 2 to c and, for each q, increasing values of z from 0 to U as

$$f_q(z) = \begin{cases} f_{q-1}(z) & \text{if } z = 0, 1, \ldots, a_q - 1; \\ \min(f_{q-1}(z), f_{q-1}(z - a_q) + p_q) & \text{if } z = a_q, \ldots, U. \end{cases}$$

The optimal solution value of formulation (5.5)-(5.7) is then given by $f_c(U)$. The time complexity of algorithm DP is $O(cU)$, which proves that TPESC with two demand nodes with identical exclusionary constraints can be solved in pseudo-polynomial time. We have shown the following:

Theorem 13. *Algorithm DP is a pseudo-polynomial time algorithm for TPESC with identical exclusionary constraints and two demand nodes.*

We now argue that it is unlikely that this algorithm can be extended to the case of three demand nodes by showing that TPESC with identical exclusionary sets and three demand nodes is strongly NP-hard.

Theorem 14. *The feasibility version of TPESC with identical exclusionary sets is strongly NP-complete, even if $|D| = 3$.*

Proof. We prove the theorem by presenting a reduction from Graph 3-colorability (see Garey & Johnson (1979)) to TPESC.

Graph 3-colorability
Input: a graph $G = (V, E)$.
Question: is G 3-colorable, that is, does there exist a coloring of the vertices of G such that two connected vertices receive different colors, and such that no more than three different colors are used?

We build an instance of TPESC by having a supply node for every vertex of V, by having a supply node for every edge of E, and by having a single dummy node d. Thus $S = V \cup E \cup \{d\}$. Each supply node corresponding to a vertex or an edge of G has $s_j = 1$, $j \in S \setminus \{d\}$, the supply corresponding to the dummy node equals $s_d = 2(|V| + |E|)$. There are three demand nodes, each having demand $d_j = |V| + |E|$. Let the two endpoints of an edge $e \in E$ be denoted by v_e and w_e, and let p_e be the supply node in S corresponding to this edge e. For each edge e in G there are three pairs of supply nodes in the set of exclusionary constraints, i.e. $F = \{(v_e, w_e), (v_e, p_e), (w_e, p_e) | e \in E\}$. Further, each supply node is connected to each demand node. This completes the description of an instance of TPESC.

Suppose that G admits a 3-coloring. We associate a different color to each of the three demand nodes. Next, we send the unit supply of each supply node corresponding to a vertex $v \in V$ to the appropriate demand node (the one with v's color in the coloring). The unit supply of a supply node corresponding to an edge $e \in E$ is sent to the demand node not receiving supply from supply nodes v_e and w_e (obviously, there is always one such node). We use the supply from the dummy node to satisfy all demand from the demand nodes exactly. Observe that we have satisfied the exclusionary constraints, and hence: we have a feasible solution to TPESC.

Suppose there is a feasible solution to TPESC. Consider a triple of supply nodes (v_e, w_e, p_e) associated to edge e. Due to the choice for F, it follows that the supply of each of these three supply nodes is sent to a unique demand node. Thus, the supply of supply nodes that correspond to adjacent vertices in G, goes to different demand nodes. Since there are three demand nodes, we have found a 3-coloring. □

5.3 TPESC with a single exclusionary set

In this section we deal with a special case of TPESC, namely the problem that arises when exactly one F-set is nonempty. As we shall see, already this restricted version is hard to approximate, even for two demand nodes. First, we sketch an application of this special case of TPESC.

When a company decides to store its items, it basically has the choice between constructing its own private warehouse and renting a public warehouse. Assuming that there are seasonal changes in the need for storage space, Ballou (1998) shows that it is advisable to make use of both options. This leaves the company with the problem of where to store what items, minimizing the total cost. One can imagine that the public warehouse imposes constraints on what items can be stored together (e.g. hazardous materials), whereas these constraints could be non-existing in a private warehouse, since this warehouse can be built specifically according to the (safety) needs of the company. This practical application boils down to a TPESC with only two demand nodes, where only one has a nonempty F-set (namely the demand node corresponding to the public warehouse).

We now show that it is difficult to find a good approximation algorithm for TPESC with a single exclusionary set.

Theorem 15. *TPESC with a single exclusionary set does not admit a polynomial-time constant-factor approximation algorithm unless $P = NP$, even if $|D| = 2$.*

Proof. We prove the theorem by presenting a reduction from Independent Set (IS) to TPESC.

Independent Set
Input: a graph $G = (V, E)$ and an integer $K \leqslant |V|$.
Question: does there exist an independent set of cardinality at least K, that is, a subset $V' \subseteq V$ with $|V'| \geqslant K$, such that no two vertices in V' are joined by an edge in E?

For each vertex $j \in V$ we construct a supply node with supply $s_j = 1$; there is an additional supply node q with supply $s_q = K$. There are two demand nodes; the first one has demand $d_1 = K$, the second one has demand $d_2 = |V|$. The cost of an edge between supply node q and the first demand node equals $c > 0$, all other edges have cost 0. The first demand node has a set of exclusionary constraints $F_1 = \{(k, l)|\ k, l \in V \wedge (k, l) \in E\}$. The second demand node has no exclusionary constraints, that is, $F_2 = \emptyset$.

We now show that the existence of a polynomial-time algorithm with a constant performance ratio for TPESC would imply $P = NP$.

Suppose that the instance of IS has a yes-answer, that is, there exists an independent set V' of cardinality at least K. In this case, given the construction of F_1, there exist K supply nodes corresponding to nodes from the set V' that satisfy the exclusionary constraints. It is now easy to see that a solution where the supply of these nodes is sent to the first demand node and where the other nodes supply the second demand node, is a feasible solution to TPESC that has zero cost.

In case that the instance of TPESC admits a zero cost solution, apparently no edge between supply node q and the first demand node is used. Hence, the demand of this node is fulfilled by K supply nodes that correspond vertices in G that form an independent set of size K.

Thus a polynomial-time algorithm with a constant performance ratio for TPESC would find a zero cost solution if one exists, and hence would be able to distinguish between the yes-instances and the no-instances of IS. □

5.4 TPESC with a fixed number of supply nodes

In this section we show that if the number of supply nodes is not part of the input, a pseudo-polynomial time algorithm exists to solve the problem. Observe that this contrasts with the case of a fixed number of demand nodes (in particular Theorem 15), where the case of two demand nodes renders a problem that does not allow a polynomial-time algorithm with a constant performance ratio (unless $P = NP$).

Theorem 16. *TPESC with a fixed number of supply nodes can be solved with a pseudo-polynomial time algorithm.*

Proof. We prove the theorem by presenting a dynamic programming algorithm for TPESC with a fixed number of supply nodes. To facilitate the exposition, let $m = |S|$, $n = |D|$, and let L be the largest number in the input. As a state in the dynamic program, we use $(f_1, f_2, \ldots, f_m)$ where f_i denotes the amount of items sent by supply node i to all demand nodes.

Observe that the number of states is bounded by $(L+1)^m$. Further we define $H_j, 1 \leqslant j \leqslant n$, as the set of states that can be reached after having fulfilled the demand of the demand nodes $1, 2, \ldots, j$, and we start with $H_0 = \{(0, 0, \ldots, 0)\}$. In iteration j, we deal with demand node j that has demand d_j, $1 \leqslant j \leqslant n$. We enumerate all possible integral ways of distributing demand d_j over the m supply nodes. Notice that we use here the fact that if a solution exists, there exists one with integral flows. Let us define E_j as the set of m-vectors that correspond to a feasible way of distributing demand d_j over the m supply nodes. In the absence of exclusionary constraints, i.e., if $F_j = \emptyset$, then

$$|E_j| = \begin{pmatrix} d_j + m - 1 \\ m - 1 \end{pmatrix}.$$

By enumerating all $\begin{pmatrix} d_j + m - 1 \\ m - 1 \end{pmatrix}$ potential ways of distributing demand d_j over the m supply nodes, and next verifying, for each way, whether it is feasible with respect to the exclusionary constraints (whose number $|F_j|$ is bounded by $\begin{pmatrix} m \\ 2 \end{pmatrix}$), we can find in $O(m^2(d_j + m)^m)$, the set E_j. Now, we can compute H_j as follows:

$$H_j = \{f + g | \ f \in H_{j-1}, g \in E_j\}.$$

States in which a value f_i exceeds s_i are omitted since they cannot lead to a feasible solution. Finally, we need to inspect whether $(s_1, s_2, \ldots, s_m) \in H_n$. If so, a solution is found, else no solution exists. The complexity of this algorithm is $O(n \cdot L^m \cdot m^2(L+m)^m)$, which, in case of a fixed m leads to a pseudo-polynomial time algorithm. Notice that when arbitrary costs c_{ij} are given, we can, by keeping track of the cost of an element of E_j, compute the cost of a state, thereby finding the cost of an optimal solution. □

It is not hard to see that Theorem 16 is best possible in the sense that one easily verifies that the existence of a polynomial-time algorithm for TPESC even with two supply nodes would imply $P = NP$. Indeed, the well-known partition problem is easily seen to be a special case of the feasibility version of TPESC with two supply nodes.

5.5 Conclusion

In this chapter, we showed that the feasibility version of TPESC is NP-complete, even if there are only two demand nodes. For the special case where all exclusionary constraints are identical in each demand node, we showed that in case of two demand nodes the feasibility version of TPESC is (weakly) NP-complete and a pseudo-polynomial time algorithm exists. In case of three demand nodes, this problem becomes strongly NP-complete. If only one single demand node has exclusionary constraints, the existence of a polynomial-time algorithm with a fixed performance ratio would imply $P = NP$, even in the case of two demand nodes. Finally, we also investigated a setting with a fixed number of supply nodes. For this special case, we showed that in case of two demand nodes the feasibility version of TPESC is (weakly) NP-complete and a pseudo-polynomial time algorithm exists. Table 5.1 gives an overview of our results.

Our results settle the complexity status of the transportation problem with exclusionary side constraints and motivate the use of heuristics and branch-and-bound approaches (Cao 1992, Sun 2002, Cao & Uebe 1995, Syarif & Gen 2003) for solving large instances.

TPESC with identical exclusionary sets		TPESC with a single exclusionary set	TPESC with a fixed number of supply nodes
$\lvert D\rvert = 2$	$\lvert D\rvert \geqslant 3$	$\lvert D\rvert \geqslant 2$	$\lvert S\rvert \geqslant 2$
Weakly NP-hard; Pseudo-pol. time algorithm	Strongly NP-hard	No polynomial-time constant-factor approximation (unless $P = NP$)	Weakly NP-hard; Pseudo-pol. time algorithm

Table 5.1: The complexity status of TPESC

Chapter 6

The matrix bid auction

This chapter focusses on a combinatorial auction where a bidder can express his preferences by means of a so-called *ordered matrix bid*. This matrix bid auction was developed by Day (2004) and allows bids on all possible subsets, although there are restrictions on what a bidder can bid for these sets. Therefore, the matrix bid auction can be classified in the restricted-preference approach (see chapter 1). In the first section, we give an overview of how this auction works and we discuss some of its properties. We elaborate on the relevance of the matrix bid auction in section 6.2. In section 6.3, we develop methods to verify whether a given matrix bid satisfies a number of properties related to microeconomic theory. Finally, in section 6.4, we investigate how a collection of arbitrary bids can be represented as a matrix bid.

6.1 Description of the matrix bid auction

The matrix bid auction is a multi-item, single-unit combinatorial auction. This means that for each item that is auctioned, only one unit of this item is available. In the matrix bid auction, each bidder must submit a strict ordering (or ranking) of the items in which he is interested. Furthermore, we assume that for each bidder, the extra value an item adds to a set is determined only by the number of higher ranked items in that set, according to the ranking of that bidder.

The ordering of the items is denoted by r_{ij}, which is item i's position in bidder j's ranking, for each $i \in G$ and $j \in B$. This ordering should be strict in the sense that for each bidder j, $r_{i_1 j} \neq r_{i_2 j}$ for any pair of distinct items i_1 and i_2. For instance, if $r_{ij} = 2$, item i is bidder j's second highest ranked item. Furthermore, each bidder j specifies values b_{ijk}, which correspond to the value the bidder is prepared to pay for item i given that it is the k-th highest ranked item in the set that bidder j is awarded. The b_{ijk} values allow to determine the value bidder j attributes to any set $S \subseteq G$. Indeed, bidder j's bid on a set S is denoted as $b_j(S)$ and can be computed as:

$$b_j(S) = \sum_{i \in S} b_{i,j,k(i,j,S)} \tag{6.1}$$

where $k(i, j, S)$ is the ranking of item i amongst the items in the set S, according to bidder j's ranking. Notice that equation (6.1) assumes that no externalities are involved, i.e. a bidder's valuation depends only on the items he wins, and not for instance on the identity of the bidders to whom the other items are allocated. The winner determination problem is, given the bids $b_j(S)$ for each set S and each bidder j, to determine which bidder is to receive which items, such that the total winning bid value is maximized. Notice that we assume that each bidder pays what he bids for the subsets he wins.

Observe that the value for index k of item i in bidder j's bid can never be higher than the rank r_{ij}. This allows us to arrange the values b_{ijk} as a lower triangular matrix for each bidder j, where the rows correspond to the items, ordered by decreasing rank and the columns correspond to values for k. Hence the name matrix bid (with order). Notice also that bidder j's ranking r_{ij} does not necessarily reflect a preference order of the items. If an item is highly ranked, this merely means that its added value to a set depends on less items than the added value of a lower ranked item. Furthermore, we make no assumption regarding the b_{ijk} values. Indeed, these values may be negative, e.g. to reflect the disposal cost of an unwanted item. Specifying a sufficiently large negative value can also keep the bidder from winning this item in the first place.

As a frivolous example, we consider the following matrix bid, where a bidder expresses his preferences for an ice cream. There are two flavors of ice cream (vanilla and banana), and also hot chocolate and strawberry sauce are available.

vanilla ice	4			
banana ice	5	2		
hot chocolate	-5	0	3	
strawberry sauce	-5	0	3	-1

Consider now the value this bidder j attributes to vanilla ice with hot chocolate. Observe that for this choice of S, vanilla ice is the highest ranked item (that is, $k(\textit{vanilla ice}, j, S) = 1$), and hot chocolate is the second highest ranked item (that is, $k(\textit{hot chocolate}, j, S) = 2$). We find using (6.1):

$$\begin{aligned} b_j(S) &= b_{\textit{vanilla ice},j,k(\textit{vanilla ice},j,S)} + b_{\textit{hot chocolate},j,k(\textit{hot chocolate},j,S)} \\ &= b_{1,j,1} + b_{3,j,2} \\ &= 4 + 0 = 4. \end{aligned}$$

Thus, this matrix bid can be interpreted as follows: bidder j feels that he needs at least one scoop of ice cream of one of the two available flavors, although he prefers banana. Indeed, no combination without ice cream will result in a positive valuation, because the bidder charges a (disposal) cost of 5 if he gets one or both toppings without ice cream. Furthermore, the bidder is not willing to pay as much for the second scoop of ice cream as for the first. The highest bid this bidder places is 9, for the combination of vanilla and banana ice with any one of the two toppings.

Despite the fact that we adopt a restriction on the preferences a bidder can express, the winner determination problem of the matrix bid auction remains NP-hard (Day 2004). In section 7.1, we elaborate on the computational complexity of this problem.

6.2 Motivation

There are three reasons for investigating a combinatorial auction with matrix bids. Shortly sketched, these reasons are: capturing structure, allowing faster computation, and finding a compact way of representing preferences. We now explore these arguments in more detail.

First, bids in any practical combinatorial auction are likely to posses some structure. In literature, we find references of both theoretical structures (see e.g. Rothkopf et al. (1998), Nisan (2000), and Leyton-Brown & Shoham (2005)) and structures in practice (see e.g. Bleischwitz & Kliewer (2005) and Goossens et al. (2007)). Capturing and understanding this structure is important, not only since it allows to develop algorithms that can be more efficient than algorithms for a general combinatorial auction, but also because it improves our understanding of various properties of an auction. The matrix bid auction, where the incremental value an item adds to a bid on a set is determined only by the number of higher ranked items in that set, imposes one such structure. Thus, the matrix bid auction offers a way of capturing structure that may be present in combinatorial auctions. Day (2004) illustrates that this structure encompasses the following six types of bidders.

Additive preference bidder: For every item i, the bidder has a price p_i.

The bidder's valuation for a set S is then $\sum_{i \in S} p_i$.

Single-minded bidder: This bidder is interested in one particular set S for which he is willing to pay a price p. These single minded bids (S, p) are also known as flat bids or atomic bids (Nisan 2000).

Nested flat bidder: This bidder is a generalization of the single-minded bidder and expresses a chain of q exclusive single-minded bids (S_1, p_1), (S_2, p_2), ... (S_q, p_q) such that $S_1 \subset S_2 \subset ... \subset S_q$.

Nested k-of bidder: The k-of bid function consists of a bid (k, S, p), which is a bid of p on any subset of S of at least size k. Multiple k-of bids (k_1, S, p_1), (k_2, S, p_2), ..., (k_q, S, p_q) on the same set S can be represented in a single matrix bid, provided that all k-values are different. This bid function is also known as the general symmetric bid function, in which the bidder specifies prices p_1, p_2, ..., p_m where p_k is the price the bidder is willing to pay for the k-th item won (see Nisan (2000) and section 1.2.3). The bidder's valuation for a set S is then $\sum_{i=1}^{|S|} p_i$.

Partition bidder: This bidder partitions the items into a number of groups of substitutes. The bidder gives a ranking of the groups and prices he is willing to pay for receiving exactly one item from each group, given that exactly one item from each higher ranked group has been received. This bid function can be generalized to accommodate an arbitrary given demand for each group of substitutes.

Add-on bidder: This bid function consists of a bid for an essential item, and extra prices the bidder is willing to pay for each number of items from a set of add-on items in which the bidder is interested.

Any auction whose bidders are from these types is a combinatorial auction with matrix bids.

Second, matrix bid auctions allow for a faster computation due to the restriction on the preferences that is assumed. Indeed, Day & Raghavan (2006) solve the winner determination problem of the matrix bid auction using a formulation based on the assignment problem (see section 7.2), and compare this with solving the winner determination problem of a general combinatorial auction using the set packing formulation (see (1.1)-(1.4)). The authors conclude that the computation time for the general combinatorial auction is higher and grows much faster than for the matrix bid auction. Moreover, they manage to solve the winner determination problem for matrix bid auctions with 72 items, 75 bidders and over 10^{23} bids, whereas for the general combinatorial auction, the largest instances that can be solved have 16 items, 25 bidders, and less than 10^9 bids. Although this comparison is somewhat distorted since it does not use a state-of-the-art method (e.g. CABOB, see Sandholm, Suri, Gilpin & Levine (2005)) to solve the winner determination problem for the general auction, it does give an indication of the size of the matrix bid auctions that can be solved in practice. We will come back to computational issues extensively in chapter 7.

Finally, the matrix bid auction also offers a compact way of representing preferences. Indeed, each bidder only needs to communicate an ordered list of m items and $\frac{m(m+1)}{2}$ matrix bid entries, which is far less than bids for each of the 2^m possible sets of items in a general combinatorial auction. We do recognize that choosing a ranking of the items and filling the matrix bid with appropriate values might not be a trivial task for the bidder. However, in section 6.4.1, we develop a procedure that recognizes whether a given collection of bids can be translated into a matrix bid. If this is not the case, the algorithm in section 6.4.2 presents a way to approximate this collection of bids by a matrix bid, in a way that does not expose the bidder to paying more than he stated for any set of items.

6.3 Recognizing matrix bid properties

Bid functions may or may not posses various micro-economic properties, which may have consequences for the complexity and/or the approximability of the winner determination problem. For instance, if bid functions are submodular (see section 6.3.3), a greedy algorithm produces a 1/2-approximation (Lehmann, Lehmann & Nisan 2001). Therefore, it is interesting to see how ordered matrix bids relate to such properties. More specifically, in this section we discuss the relationship between the bid function implied by a matrix bid and micro-economic concepts as free disposal (6.3.1), complement freeness (6.3.2), decreasing marginal valuations (6.3.3), and the gross substitutes property (6.3.4). In particular, we show how to verify efficiently whether a given matrix bid satisfies each of these properties. Since this section deals only with the bid function of a single bidder, the index j that is used to indicate the bidder will be dropped. In literature, many of the economic concepts discussed in this section are in terms of a valuation function. Although for some auctions (e.g. the VCG auction), it has been shown that it is in the bidder's best interest to bid his true valuation, in general, a bidder's bid and his valuation need not be identical. Indeed, strategic considerations may motivate a bidder to express bids that differ considerably from his valuation. Nevertheless, in this section, we ignore this issue and assume equivalence between the notions bid function and valuation function. This is common practice in studies on bidding languages (see e.g. Nisan (2005)). We refer to Gul & Stacchetti (1999) and Cramton et al. (2005) for the definitions used in this section.

6.3.1 Free disposal

In microeconomics, it is often assumed that agents prefer more to less. In the context of an auction, this means that a bidder is always willing to receive one or more items for free (see also section 3.2). Consequently, a seller will never get stuck with unsold items and can therefore dispose of any number of items at no cost. Free disposal is a very common assumption in literature on combinatorial auctions (Lehmann et al. 2001, Nisan 2000) and can be defined as follows.

Definition 1. *A bid function b satisfies the free disposal property if*

$$b(S) \leqslant b(T) \qquad \forall S \subseteq T \subseteq G. \tag{6.2}$$

Notice that this definition is equivalent with the definition of a monotone non-decreasing function. Alternatively stated, this definition implies that disposing an item from a set cannot increase the value of the resulting set.

In combinatorial auctions where a bidder does not communicate bids on every possible subset of items, but rather only on a limited number of subsets of his liking, allocating all items may be problematic for the auctioneer. In its strictest sense, the lack of free disposal would mean that buyers do not accept anything extra beyond what they bid on. Using this interpretation, even finding a solution to the winner determination problem where all items are allocated is NP-complete (Sandholm, Suri, Gilpin & Levine 2002). However, many other approaches allow the auctioneer to allocate all items, without disposal cost. Nisan (2005) assumes that bids of each bidder satisfy the free disposal property. Moreover, if a bidder did not express a bid on a set S, the auctioneer can construct a new bid $b(S)$ equal to the highest bid over all subsets of S. Obviously, the newly created bids also satisfy the free disposal property. A similar approach is followed by Leyton-Brown, Shoham & Tennenholtz (2000), since they allow the auctioneer to create additional bids with value zero for any subset of items, which can then be combined with any of the bids expressed by the bidders (which also satisfy the free disposal property).

The concept of free disposal is, however, also relevant in a combinatorial auction where bidders do express bids on every possible subset, as in the matrix bid auction. In this case, assuming bid functions that satisfy free disposal guarantees the existence of a total winning bid value maximizing allocation in which all items are awarded to some bidder.

Obviously, not every matrix bid satisfies the free disposal property. For instance, the matrix bid that was used in section 6.1 does not allow free disposal, since $b(\textit{hot chocolate}) < b(\varnothing)$. The matrix framework indeed allows the bidder to take into account a disposal cost which can vary across the items and may lead to one or more sets with a negative valuation. However, imposing that each entry in the matrix bid is non-negative is not sufficient to attain the free disposal property. This is illustrated by the matrix bid below, as $b(y) = 3 > 2 = b(x, y)$.

$$\begin{array}{r|ll} \text{item } x & 1 & \\ \text{item } y & 3 & 1 \end{array}$$

Verifying free disposal can be done in polynomial time for a given matrix bid, as witnessed by the following theorem.

Theorem 17. *Verifying whether a matrix bid b satisfies the free disposal property can be done in polynomial time.*

Proof. We will show that solving a shortest path problem on an acyclic graph involving $O(m^3)$ nodes and $O(m^4)$ arcs determines whether a matrix bid b satisfies the free disposal property (6.2). The graph can be described as follows.

The graph contains a source and a sink, and nodes indexed by (i, s, t). The index i refers to item i and ranges from 1 to m. The index t ranges from 1 to r_i, while s ranges from 0 to t. There are arcs from each node (i, s, t) to $(i', s, t + 1)$ and to $(i', s + 1, t + 1)$, for all items i' ranked lower than i (recall that we consider a single bidder). Furthermore, there is an arc from the source to each node $(i, 0, 1)$ and $(i, 1, 1)$, and there is an arc from each node (except the source) to the sink. Let the cost on the arc from (i, s, t)

to $(i', s, t+1)$ be equal to $b_{i',t+1}$ and let the cost on the arc from (i, s, t) to $(i', s+1, t+1)$ be equal to $b_{i',t+1} - b_{i',s+1}$. Analogously, the arcs from the source to each node $(i, 0, 1)$ and $(i, 1, 1)$ have a cost equal to $b_{i,1}$ and zero respectively. Arcs to the sink have a cost equal to zero. This completes the description of the graph. Notice that this graph is acyclic and contains a number of nodes and arcs that is polynomial in the number of items ($O(m^3)$ and $O(m^4)$ respectively). Figure 6.1 illustrates this graph; arcs with no indication of their cost have a cost equal to 0.

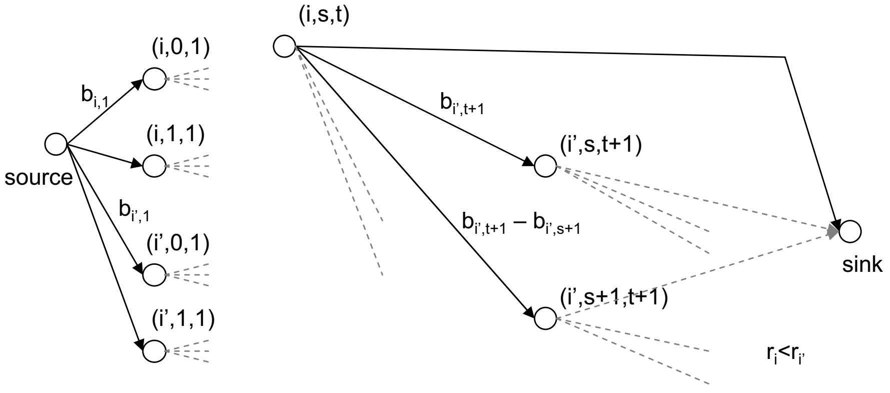

Figure 6.1: Graph used to verify free disposal

The graph described above should be interpreted as follows. Each node (i, s, t) corresponds to a state where s and t items ranked at least as high as item i are present in set S and set $T \supseteq S$ respectively. Selecting an arc from (i, s, t) to $(i', s, t+1)$ corresponds to adding item i' to set T as the $(t+1)$-th best item, but not to S, whereas an arc from (i, s, t) to $(i', s+1, t+1)$ corresponds to adding item i' to both set S and set T, as the $(s+1)$-th and $(t+1)$-th best item respectively. In this way, each path from source to sink determines sets S and T, and, vice versa, there is a path from the source to the sink for each possible S and T. Notice that the arcs are such that S will always be a subset of T.

We now show that the length of a path from source to sink in this graph equals $b(T) - b(S)$. Each path from source to sink consists of two types of arcs: those arcs that add items i to the set T (and not to S), and those arcs that add items i to both S and T. The former arcs give rise to the term $\sum_{i \in T \setminus S} b_{i,k(i,T)}$, whereas the latter arcs give rise to the term $\sum_{i \in S \subseteq T}(b_{i,k(i,T)} - b_{i,k(i,S)})$. Recall from section 6.1 that $k(i, A)$ denotes the rank of item i in the set A. Thus, the length of the path equals

$$\begin{aligned} \sum_{i \in T \setminus S} b_{i,k(i,T)} + \sum_{i \in S \subseteq T} (b_{i,k(i,T)} - b_{i,k(i,S)}) &= \sum_{i \in T} b_{i,k(i,T)} - \sum_{i \in S} b_{i,k(i,S)} \\ &= b(T) - b(S). \end{aligned}$$

Thus, verifying the free disposal property (see definition 1) for a given matrix bid can be done by solving a shortest path problem in this graph, which takes polynomial time. Concluding, a shortest path in our graph with nonnegative length is equivalent to free disposal. □

6.3.2 Complement free

Although the difficulty to deal with complementarity or substitution effects in a bidder's valuation in a classic sequential auction is a major motivation for researching combinatorial auctions in the first place, assuming the absence of complementarities is quite common in microeconomic theory. Lehmann et al. (2001) state that "in most of microeconomic theory, the consumers are assumed to exhibit diminishing marginal utilities". In their work, they assume that the valuation of a union of disjoint sets is never higher than the sum of the valuations of the individual sets. This notion can be formalized as follows and is also known as subadditivity.

Definition 2. *A bid function b is complement free (or subadditive) if*

$$b(S \cup T) \leqslant b(S) + b(T) \qquad \forall S, T : S \cap T = \varnothing. \tag{6.3}$$

Although the winner determination problem for bidders with a complement free bid function remains NP-hard (Lehmann et al. 2001), there exists a polynomial time algorithm that finds a $O(\frac{1}{\log m})$-approximation if, given external prices for all items, a bidder can determine in polynomial time for which set his valuation b exceeds the sum of the prices of the items in that set the most (Dobzinski, Nisan & Schapira 2005). If a bidder can only determine his valuation for a given set in polynomial time, then the approximation ratio decreases to $O(m^{-1/2})$ (Dobzinski et al. 2005).

A sufficient condition to have a complement free matrix bid is that the b_{ik} values are non-increasing in the rows (i.e. $b_{ik} \leqslant b_{i,k+1}$ $\forall i, k$). Indeed, for each item i, the valuation of the union of two sets will only make use of b_{ik} values with a value of index k at least as high as the value used in the valuation of the individual sets. Having non-increasing b_{ik} values is not a necessary condition though, as is illustrated by the following example of a complement free bid function.

item x	0		
item y	2	1	
item z	2	1	2

We now show how we can verify in polynomial time whether a matrix bid is complement free.

Theorem 18. *Verifying whether a matrix bid b satisfies the complement free property can be done in polynomial time.*

Proof. Given definition 2, it suffices to establish the existence of sets of items S and T, such that $S \cap T = \varnothing$ and $b(S) + b(T) - b(S \cup T) < 0$, to find out whether a matrix bid b is not complement free. We show that this can be done by solving a shortest path problem, by adapting the construction described in Theorem 17.

Consider a graph containing a source, a sink, and nodes, indexed by (i, s, t). The index i refers to item i, with $1 \leqslant i \leqslant m$. Both indices s and t range from 0 to r_i, insofar $0 < s+t \leqslant r_i$. There are arcs from each node (i, s, t) to $(i', s+1, t)$ and to $(i', s, t+1)$, for all items i' ranked lower than i. Furthermore, there is an arc from the source to each node $(i, 1, 0)$ and $(i, 0, 1)$, and there is an arc from each node (except the source) to the sink. Let the cost of the arc from (i, s, t) to $(i', s+1, t)$ be equal to $b_{i',s+1} - b_{i',s+t+1}$ and let the cost of the arc to $(i', s, t+1)$ be equal to $b_{i',t+1} - b_{i',s+t+1}$. The arcs from the source to each node $(i, 1, 0)$ and $(i, 0, 1)$, and also all arcs to the sink have a cost equal to zero. Notice that this graph is acyclic. Figure 6.2 illustrates this graph; arcs with no indication of their cost have a cost equal to 0.

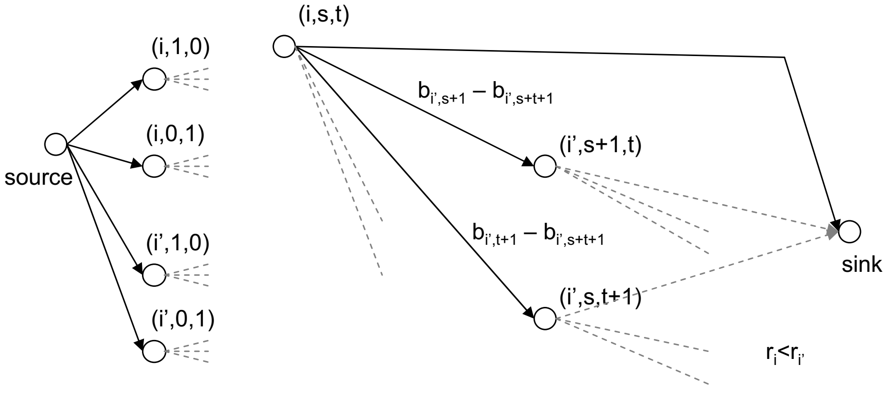

Figure 6.2: Graph used to verify complement freeness

Each node (i, s, t) in the graph represents the state where s and t items ranked at least as high as item i are present in sets S and T respectively. Each path from source to sink determines what items are to be added to sets S and T, and the arcs are such that these sets will always be disjoint. If such a path contains an arc from (i, s, t) to $(i', s+1, t)$, this corresponds to item i' being present in set S as the $(s+1)$-th best item. If an arc from (i, s, t) to $(i', s, t+1)$ is present in the path, this means that item i' is in set T as the $(t+1)$-th best item. We now show that the length of a path from source to sink in this graph equals $b(S)+b(T)-b(S\cup T)$. Each path from source to sink consists of two types of arcs: those arcs that add item an item i to the set S, and those arcs that add an item i to T. Given the choice of the costs of the arcs, the length of a path equals

$$\begin{aligned}\sum_{i\in S}&(b_{i,k(i,S)}-b_{i,k(i,S\cup T)})+\sum_{i\in T}(b_{i,k(i,T)}-b_{i,k(i,S\cup T)})\\ &=\sum_{i\in S}b_{i,k(i,S)}+\sum_{i\in T}b_{i,k(i,T)}-\sum_{i\in S\cup T}b_{i,k(i,S\cup T)}\\ &=b(S)+b(T)-b(S\cup T).\end{aligned}$$

Thus, if a shortest path in this graph has a non-negative length, then the matrix bid b is complement free, and vice versa. Since the graph is acyclic and contains a number of nodes and arcs that is polynomial in the number of items ($O(m^3)$ and $O(m^4)$ respectively), verifying whether a matrix bid is complement free can be done in polynomial time. □

Complement free valuations find their natural counterpart in substitute free valuations, for which $b(S\cup T)\geqslant b(S)+b(T)$ for all disjoint sets S and T. This property is also known as superadditivity. The set packing formulation of the winner determination problem (see (1.1)-(1.4)) can be written more succinctly if all bid functions are superadditive (de Vries & Vohra 2003). In this case, there is no need to prevent a bidder from winning multiple bids. Furthermore, the auctioneer only needs to take into account the highest bid for each set.

Notice that having b_{ik} values that are non-decreasing in the rows is a sufficient condition for a matrix bid to be superadditive. Non-decreasing b_{ik} values in the rows is, however, not a necessary condition, which is illustrated by the matrix bid below.

item x	0		
item y	1	2	
item z	2	3	2

Verifying whether a matrix bid satisfies the substitute free property can also be done in polynomial time. Using the same graph as in theorem 18, a longest path with non-positive length implies superadditivity of the given matrix bid. Since this graph has no cycles, its longest path can be found in polynomial time.

Corollary 1. *Verifying whether a matrix bid b satisfies the substitute free property can be done in polynomial time.*

6.3.3 Decreasing marginal valuations

In many practical applications, and also in most of microeconomic theory, it is assumed that the more items an agent has, the less he values an extra item. This concept is called decreasing marginal valuations. It is also an assumption in chapters 2 to 4, where a buyer expects a discount from its suppliers if a larger volume was purchased.

Definition 3. *A bid function b has decreasing marginal valuations if*

$$b(T \cup \{x\}) - b(T) \leqslant b(S \cup \{x\}) - b(S) \qquad \forall S \subseteq T, x \in G. \tag{6.4}$$

Moulin (1988) showed that a (bid) function has decreasing marginal valuations if and only if it is submodular.

Definition 4. *A bid function b is submodular if*

$$b(S \cup T) + b(S \cap T) \leqslant b(S) + b(T) \qquad \forall S, T \subseteq G. \tag{6.5}$$

Lehmann et al. (2001) show that a valuation where the items have decreasing marginal valuations is also complement free (assuming that this valuation function satisfies free disposal and normalization). The authors also provide an example that illustrates that the converse is not true. Indeed, a complement free valuation function may still have so-called hidden complementarities. When we consider a bidder's valuation for a set of items A, given that this bidder already acquired some set of items W ($W \cap A = \emptyset$), complementarities may still arise. In other words, even if a bid function b is complement free, this is not necessarily the case for marginal bids b', defined by $b'(A) = b(A|W) = b(A \cup W) - b(W)$. If we want to enforce that the marginal bids are complement free as well, the bid function b is required to be submodular (Lehmann et al. 2001). This phenomenon may play a role in an auction with multiple rounds.

Lehmann et al. (2001) also show that the winner determination problem for bidders with a submodular bid function remains NP-hard, but that a greedy algorithm produces a 1/2-approximation. This algorithm simply assigns the items one by one (in no particular order) to the bidder with the highest marginal value for that item, given the other items that bidder already acquired. The following example with two items (x and y) and two bidders (A and B) shows that this approximation is tight, even for submodular matrix bids.

bidder A			bidder B		
item x	1		item x	1	
item y	1	0	item y	0	0

The optimal total winning bid value for the auctioneer in this auction is 2, by allocating item x to bidder B and item y to bidder A. However, the greedy algorithm can generate a total winning bid value of 1, by starting with allocating item x arbitrarily to bidder A and ending up with marginal bids of zero for item y. Khot, Lipton, Markakis & Mehta (2005) show that if each bidder can determine his valuation for a given set in polynomial time and if this valuation function is submodular, it is NP-hard to approximate the optimal solution by a factor better than $\frac{e-1}{e}$. This result assumes that, given external prices for all items, bidders cannot determine in polynomial time (in the number of items and bidders) for which set their valuation exceeds the sum of the prices of the items in that set the most.

Day (2004) suggests that a matrix bid with b_{ik} values that are non-increasing in both the rows and the columns represents a bid function with decreasing marginal valuations. Notice from the example below that this is not necessarily the case (with $S = \{x, z\}$ and $T = \{y, z\}$).

item x	7		
item y	6	5	
item z	5	1	0

Furthermore, not all bid functions with decreasing marginal valuations can be represented as a matrix bid, as can be easily verified for the submodular bid function b that produces the following bids on each subset of the item set $\{x, y, z\}$: $b(\{\}) = 0$, $b(\{x\}) = 1$, $b(\{y\}) = 2$, $b(\{z\}) = 3$, $b(\{x, y\}) = 3$, $b(\{y, z\}) = 3$, $b(\{x, z\}) = 3$ and $b(\{x, y, z\}) = 3$.

We can, however, verify whether a matrix bid represents a valuation function with decreasing marginal valuations in polynomial time.

Theorem 19. *Verifying whether a matrix bid b has decreasing marginal valuations can be done in polynomial time.*

Proof. Using the equivalence result by Moulin (1988), it is sufficient to establish the existence of sets of items S and T, for which $b(S) + b(T) - b(S \cup T) - b(S \cap T) < 0$, in order to find out whether a matrix bid is not submodular. We show that the existence of such sets can be verified by solving a shortest path problem, again by adapting the construction described in Theorem 17.

Consider a graph containing a source, a sink, and nodes, indexed by (i, s, t, c). The index i refers to item i, with $1 \leqslant i \leqslant m$. The indices s, t and c range from 0 to r_i, though no nodes are needed if both s and t are 0.

There are arcs from each node (i, s, t, c) to $(i', s + 1, t, c)$, $(i', s, t + 1, c)$ and $(i', s + 1, t + 1, c + 1)$, for all items i' ranked lower than i. Furthermore, there are arcs from the source to each node $(i, 1, 0, 0)$, $(i, 0, 1, 0)$ and $(i, 1, 1, 1)$, and there are arcs from each node but the source to the sink. Let the cost on the arc from (i, s, t, c) to $(i', s + 1, t, c)$ be equal to $b_{i',s+1} - b_{i',s+t-c+1}$ and let the cost on the arc to $(i', s, t + 1, c)$ be equal to $b_{i',t+1} - b_{i',s+t-c+1}$. The arcs from (i, s, t, c) to $(i', s + 1, t + 1, c + 1)$ have a cost equal to $b_{i',s+1} + b_{i',t+1} - b_{i',s+t-c+1} - b_{i',c+1}$. The arcs from the source to each node $(i, 1, 0, 0)$, $(i, 0, 1, 0)$ and $(i, 1, 1, 1)$, and also all arcs to the sink have a cost equal to zero. Figure 6.3 illustrates this acyclic graph; arcs with no indication of their cost have a cost equal to 0.

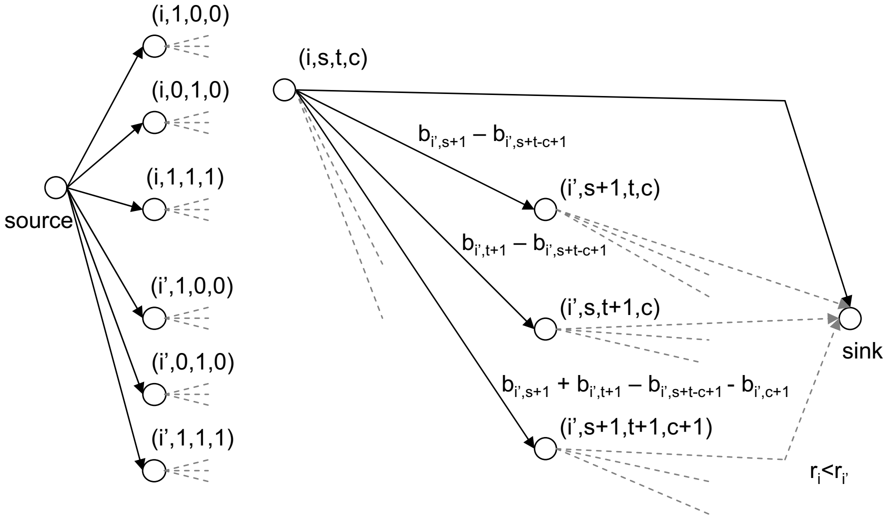

Figure 6.3: Graph used to verify decreasing marginal valuations

The graph should be interpreted as follows. Each node (i, s, t, c) stands for a state where s, t, and c items ranked at least as high as item i that are in sets S, T, and $S \cap T$ respectively. Each path from source to sink determines what items are to be added to sets S and T, and there is a path from source to sink for each possible S and T. If the arc from (i,s,t,c) to $(i',s+1,t,c)$ is included in the path, this means that item i' is in set S as the $(s+1)$-th best item, whereas the arc to $(i',s,t+1,c)$ corresponds to item i' being in set T, as the $(t+1)$-th best item. The arc from (i,s,t,c) to $(i',s+1,t+1,c+1)$ corresponds to adding item i' to both S and T, where $c+1$ is the number of items in $S\cap T$ ranked at least as high as i'.

We now show that the length of a path from source to sink corresponds to $b(S)+b(T)-b(S\cup T)-b(S\cap T)$. Each path consists of three types of arcs: those arcs that add an item i to S and not to T, those that add an item i to T and not to S, and those that add an item i to both S and T. Given the choice of the costs of these arcs, the length of the path equals

$$\begin{aligned}\sum_{i\in S\setminus T}&(b_{i,k(i,S)}-b_{i,k(i,S\cup T)})+\sum_{i\in T\setminus S}(b_{i,k(i,T)}-b_{i,k(i,S\cup T)})\\&+\sum_{i\in S\cap T}(b_{i,k(i,S)}+b_{i,k(i,T)}-b_{i,k(i,S\cup T)}-b_{i,k(i,S\cap T)})\\&=\sum_{i\in S}b_{i,k(i,S)}+\sum_{i\in T}b_{i,k(i,T)}-\sum_{i\in S\cup T}b_{i,k(i,S\cup T)}-\sum_{i\in S\cap T}b_{i,k(i,S\cap T)}\\&=b(S)+b(T)-b(S\cup T)-b(S\cap T).\end{aligned}$$

Thus, if the shortest path in this graph has a non-negative length, then the matrix bid has decreasing marginal valuations, and vice versa. Since the graph contains a number of nodes and arcs that is polynomial in the number of items ($O(m^4)$ and $O(m^5)$ respectively), it is clear that verifying whether a matrix bid b has decreasing marginal valuations can be done in polynomial time. □

If a bid function b satisfies the property that $b(S\cup T)+b(S\cap T)\geqslant b(S)+b(T)$ for all sets S and T, we call b supermodular, or, equivalently, b is said to have increasing marginal valuations. It is pointed out by de Vries & Vohra (2003) that if there are only two bid functions a bidder can have, both of them non-decreasing, integer valued and supermodular, then the corresponding winner determination problem can be solved in polynomial time.

Notice that having b_{ik} values that are non-decreasing in the rows is no sufficient condition for a matrix bid to be supermodular. Choosing $S = \{x, z\}$ and $T = \{y, z\}$ in the matrix bid below illustrates this.

item x	1		
item y	1	2	
item z	0	5	6

The same graph as described in theorem 19 can be used to verify whether a matrix bid is supermodular. Indeed, a longest path from source to sink with a non-positive length implies that the matrix bid is supermodular. Since the graph is acyclic, its longest path can be found in polynomial time.

Corollary 2. *Verifying whether a matrix bid b has increasing marginal valuations can be done in polynomial time.*

6.3.4 Gross substitutes property

The gross substitutes property was introduced by Kelso & Crawford (1982) in the context of a labor market, and applied to auctions by e.g. Bevia, Quinzii & Silva (1999), and Bikhchandani & Mamer (1997). The gross substitutes property departs from a price vector p containing prices that are to be paid for each item i. Given a valuation function b, we can define the demand set $D(p)$ of the corresponding bidder given the current price vector p as

$$D(p) = \{\underset{S \subseteq G}{\operatorname{argmax}}(b(S) - \sum_{i \in S} p_i)\}. \tag{6.6}$$

The gross substitutes property requires that a bidder will continue to demand items for which the price did not rise, when other items have become more expensive. This condition can be defined more formally as follows.

Definition 5. *A bid function b satisfies the gross substitutes property if for all price vectors $p \leqslant q$ (according to a point-wise comparison) and all sets $S \in D(p)$, there exists a set $T \in D(q)$ such that $\{i \in S : p_i = q_i\} \subseteq T$.*

The gross substitutes property is stronger than submodularity, since Gul & Stacchetti (1999) show that each bid function that satisfies the gross substitutes condition is also submodular, whereas the converse is not true. Gul & Stacchetti (1999) also prove that in an auction where each bidder has a bid function that satisfies the gross substitutes property, there exists a price vector and an allocation such that every bidder receives a set of items that is in his demand set given these prices. This situation is known as a Walrasian equilibrium. Kelso & Crawford (1982) develop a fully polynomial approximation scheme for finding this Walrasian equilibrium (see also Nisan & Segal (2006)).

Assuming that a bid function b has the gross substitutes property leads to a number of interesting results. Indeed, the LP-relaxation of the set packing formulation (see (1.1)-(1.4)) for the winner determination problem of a combinatorial auction where all bidders have bid functions that satisfy the gross substitutes condition has an integral solution (Kelso & Crawford 1982, Bikhchandani, de Vries, Schummer & Vohra 2002). Furthermore, Murota & Tamura (2003) and Fujishige & Yang (2003) show that given a valuation function that satisfies the gross substitutes property and a price vector, the bidder's demand set can be computed in polynomial time. Using the equivalence of separation and optimization (Grötschel, Lovász & Schrijver 1981), it follows that in this setting, the winner determination problem can be solved in polynomial time. Ausubel & Milgrom (2005) show that if every bidder has gross substitutes bid functions, the

Vickrey-Clarke-Groves auction does not suffer from a number weaknesses as e.g. vulnerability to the use of multiple bidding identities by a single bidder.

The fact that the definition of the gross substitutes property is based on prices that should be paid for each item is somewhat awkward, since a matrix bid specifies bids, and the price for a set of items is simply the winning bid for that set. However, Reijnierse, Potters & van Gellekom (2002) formulated the following equivalent characterization of the gross substitutes property, which is independent of prices. A bid function b satisfies the gross substitutes property if for all $S \subseteq G$ and $x, y, z \in G$ the following conditions hold:

$$b(S \cup \{x, y\}) - b(S \cup \{x\}) \leqslant b(S \cup \{y\}) - b(S), \text{ and} \tag{6.7}$$

$$\begin{aligned} & b(S \cup \{x, y\}) + b(S \cup \{z\}) \\ & \qquad \leqslant \max(b(S \cup \{x, z\}) + b(S \cup \{y\}), b(S \cup \{y, z\}) + b(S \cup \{x\})). \end{aligned} \tag{6.8}$$

Reijnierse et al. (2002) also show that it can be checked whether a bid function satisfies the gross substitutes property in a time complexity of $O(K \log^3(K))$ where K is the number of subsets of G, which equals 2^m. For matrix bids, however, this can be done in a time which is polynomial in the number of items, as witnessed by the following theorem.

Theorem 20. *Verifying whether a matrix bid b satisfies the gross substitutes property can be done in polynomial time.*

Proof. From the equivalence result by Reijnierse et al. (2002) it follows that a matrix bid b has the gross substitutes property if and only if conditions (6.7) and (6.8) are satisfied. Moulin (1988) shows that condition (6.7) is in fact equivalent with submodularity. According to theorem 19, checking whether a matrix bid b is submodular can be done in polynomial time. As for condition (6.8), it suffices to consider the setting where $x, y, z \notin S$. Indeed, the matrix bid auction is a single-unit auction, implying $b(S \cup \{x\}) = b(S)$

if $x \in S$. Using this, it is trivial to see that condition (6.8) where any subset of $\{x, y, z\}$ is in S, is satisfied for any submodular matrix bid. In the remainder of this proof, we will show that solving a shortest path problem on an acyclic graph suffices to check whether a matrix bid b satisfies condition (6.8) with $x, y, z \notin S$. We will first assume that $r_x < r_y < r_z$ (i.e. item x is ranked higher than item y, which is ranked higher than item z) and afterwards we will discuss the other settings.

First, we show that for the setting with $r_x < r_y < r_z$, the following is true:

$$b(S \cup \{x, z\}) + b(S \cup \{y\}) = b(S \cup \{y, z\}) + b(S \cup \{x\}). \qquad (6.9)$$

Indeed, all items i in S that are ranked higher than item x contribute $b(i, k(i, S))$ to each of the four terms in equation (6.9), where $k(i, S)$ is the rank i has amongst the items in S. Item x adds $b(x, k(x, S \cup \{x, z\}))$ to the left-hand side of equation (6.9), which equals $b(x, k(x, S \cup \{x\}))$, added to the right-hand side. Items $i \in S$ that are ranked between x and y contribute $b(i, k(i, S)) + b(i, k(i, S) + 1)$ to both sides of the equation. Items $i \in S$ ranked between y and z contribute $2b(i, k(i, S) + 1)$ to both the left-hand and right-hand side of equation (6.9). Furthermore, also item y adds equal amounts to both sides of the equation, namely $b(y, k(y, S \cup \{y\}))$ and $b(y, k(y, S \cup \{y, z\}))$. The same goes for item z, adding the equal terms $b(z, k(z, S \cup \{x, z\}))$ and $b(z, k(z, S \cup \{y, z\}))$ to the left-hand and the right-hand side respectively. Finally, items $i \in S$ ranked lower than z add $b(i, k(i, S) + 2) + b(i, k(i, S) + 1)$ to both sides of the equation. Using this result, condition (6.8) can be reformulated as

$$b(S \cup \{x, y\}) + b(S \cup \{z\}) \leqslant b(S \cup \{x, z\}) + b(S \cup \{y\}). \qquad (6.10)$$

Consider a graph containing a source, a sink and nodes, indexed by (i, s, q). The index i refers to item i and ranges from 1 to m, whereas s ranges from 0 to r_i and q ranges from 0 to 3. There are arcs from each node (i, s, q)

Table 6.1: Cost on the arcs from (i,s,q) to $(i',s+1,q)$ and to $(i',s,q+1)$, depending on the value for q.

	$(i,s,q) \to (i',s+1,q)$	$(i,s,q) \to (i',s,q+1)$
$q=0$	$b_{i',s+1} + b_{i',s+1} - b_{i',s+1} - b_{i',s+1} = 0$	$b_{i',s+1} - b_{i',s+1} = 0$
$q=1$	$b_{i',s+2} + b_{i',s+1} - b_{i',s+2} - b_{i',s+1} = 0$	$b_{i',s+2} - b_{i',s+1}$
$q=2$	$b_{i',s+3} + b_{i',s+1} - b_{i',s+2} - b_{i',s+2}$	$b_{i',s+1} - b_{i',s+2}$
$q=3$	$b_{i',s+3} + b_{i',s+2} - b_{i',s+3} - b_{i',s+2} = 0$	(no such arc exists)

to $(i',s+1,q)$ and to $(i',s,q+1)$, for all items i' ranked lower than i and insofar $q+1 \leqslant 3$. Furthermore, there are arcs from the source to each node $(i,1,0)$ and $(i,0,1)$, and there are arcs from each node $(i,s,3)$ to the sink. Depending on the value for q, the arc from (i,s,q) to $(i',s+1,q)$ and the arc from (i,s,q) to $(i',s,q+1)$ have costs as presented in table 6.1. The arcs from the source to each node $(i,1,0)$, and$(i,0,1)$, and also all arcs to the sink have a cost equal to zero.

The graph can be interpreted as follows. Each node (i,s,q) represents a state where s items ranked at least as high as item i are in set S. The index q keeps track of how many of the items x, y, and z are ranked at least as high as item i, and should be understood as follows.

$$\begin{aligned} q=0: &\quad r_x > r_i \\ q=1: &\quad r_y > r_i > r_x \\ q=2: &\quad r_z > r_i > r_y \\ q=3: &\quad r_i > r_z \end{aligned}$$

Each path from source to sink determines which items are to be added to set S and which items are selected to play the role of x, y and z. If the path contains an arc from (i,s,q) or from the source to $(i',s+1,q)$, this indicates that item i' is added to the set S, as the $(s+1)$-th highest ranked item. An arc from (i,s,q) or from the source to $(i',s,q+1)$ indicates that item i' is selected as item x, y or z, for $q=0$, $q=1$, or $q=2$ respectively. The fact that there are only arcs from nodes $(i,s,3)$ to the sink, guarantees that items x, y and z are selected in each path from the source to the sink.

We now show that the length of a path from source to sink in this graph equals $b(S \cup \{x, y\}) + b(S \cup \{z\}) - b(S \cup \{x, z\}) - b(S \cup \{y\})$. Each path from source to sink has exactly one arc that selects an item x (namely the arc where q increases from 0 to 1), one arc that selects an item y (q increases from 1 to 2), and one arc that selects an item z (q increases from 2 to 3). The other arcs in the path can be divided into four types: those arcs that add an item i ranked higher than x to the set S, those arcs that add an item i ranked between x and y to S, those arcs that add an item i ranked between y and z to S, and those arcs that add an item i ranked lower than z. From the costs in table 6.1, it follows that the length of a path from source to sink equals

$$
\begin{aligned}
&\sum_{i \in S: r_x > r_i} (b_{i,k(i,S\cup\{x,y\})} + b_{i,k(i,S\cup\{z\})} - b_{i,k(i,S\cup\{x,z\})} - b_{i,k(i,S\cup\{y\})}) \\
&+ \sum_{i \in S: r_y > r_i > r_x} (b_{i,k(i,S\cup\{x,y\})} + b_{i,k(i,S\cup\{z\})} - b_{i,k(i,S\cup\{x,z\})} - b_{i,k(i,S\cup\{y\})}) \\
&+ \sum_{i \in S: r_z > r_i > r_y} (b_{i,k(i,S\cup\{x,y\})} + b_{i,k(i,S\cup\{z\})} - b_{i,k(i,S\cup\{x,z\})} - b_{i,k(i,S\cup\{y\})}) \\
&+ \sum_{i \in S: r_i > r_z} (b_{i,k(i,S\cup\{x,y\})} + b_{i,k(i,S\cup\{z\})} - b_{i,k(i,S\cup\{x,z\})} - b_{i,k(i,S\cup\{y\})}) \\
&+ b_{x,k(x,S\cup\{x,y\})} - b_{x,k(x,S\cup\{x,z\})} + b_{y,k(y,S\cup\{x,y\})} - b_{y,k(y,S\cup\{y\})} \\
&+ b_{z,k(z,S\cup\{z\})} - b_{z,k(z,S\cup\{x,z\})} \\
&= \sum_{i \in S\cup\{x,y\}} b_{i,k(i,S\cup\{x,y\})} + \sum_{i \in S\cup\{z\}} b_{i,k(i,S\cup\{z\})} \\
&- \sum_{i \in S\cup\{x,z\}} b_{i,k(i,S\cup\{x,z\})} - \sum_{i \in S\cup\{y\}} b_{i,k(i,S\cup\{y\})} \\
&= b(S \cup \{x, y\}) + b(S \cup \{z\}) - b(S \cup \{x, z\}) - b(S \cup \{y\}).
\end{aligned}
$$

Thus, if this graph has a shortest path with non-negative length, then condition (6.8) is satisfied for every S and every $x, y, z \notin S$ such that $r_x < r_y < r_z$. Moreover, a similar reasoning can be used to prove that this result is also valid if $r_y < r_x < r_z$.

We now show that for the setting with $r_x < r_z < r_y$, condition (6.8) is satisfied for any matrix bid, since

$$b(S \cup \{x, y\}) + b(S \cup \{z\}) = b(S \cup \{y, z\}) + b(S \cup \{x\}). \tag{6.11}$$

Observe that all items i in S that are ranked higher than item x contribute $b(i, k(i, S))$ to each of the terms in equation (6.11). Item x adds $b(x, k(x, S \cup \{x, y\}))$ to the left-hand side of equation (6.11), which equals $b(x, k(x, S \cup \{x\}))$, added to the right-hand side. Items $i \in S$ that are ranked between x and z contribute $b(i, k(i, S)) + b(i, k(i, S) + 1)$ to both sides of the equation. Also item z adds an equal to term to both sides of the equation, namely $b(z, k(z, S \cup \{z\}))$ and $b(z, k(z, S \cup \{y, z\}))$ to the left-hand and the right-hand side respectively. Items $i \in S$ ranked between z and y contribute $2b(i, k(i, S) + 1)$ to both the left-hand and right-hand side of equation (6.11). Furthermore, also item y adds equal amounts to both sides of the equation, namely $b(y, k(y, S \cup \{x, y\}))$ and $b(y, k(y, S \cup \{y, z\}))$. Finally, items $i \in S$ ranked lower than y add $b(i, k(i, S) + 2) + b(i, k(i, S) + 1)$ to both sides of the equation. A similar reasoning can be used to show that equality (6.11) is also valid for a setting with $r_z < r_x < r_y$. Obviously, condition (6.8) is satisfied for any matrix bid that satisfies equality (6.11).

An analogous proof can be developed to show that condition (6.8) is also always satisfied for the settings with $r_y < r_z < r_x$ or $r_z < r_y < r_x$, since

$$b(S \cup \{x, y\}) + b(S \cup \{z\}) = b(S \cup \{x, z\}) + b(S \cup \{y\}). \tag{6.12}$$

We can conclude that it can be verified in polynomial time whether a matrix bid satisfies conditions (6.7) and (6.8) by solving a shortest path problem on two graphs, with a number of nodes and arcs that is polynomial in the number of items ($O(m^4)$ and $O(m^5)$, and $O(m^2)$ and $O(m^3)$ respectively). If the shortest paths in both graphs have a non-negative length, then the matrix bid has the gross substitutes property. Otherwise, if one of the shortest paths has a negative length, the matrix bid does not have the gross substitutes property. □

6.4 Expressing arbitrary bids as a matrix bid

The matrix bid auction fits in the scope of the restricted preference approach, which accepts that bidders cannot fully express their preferences in order to solve the winner determination problem more efficiently. In section 6.4.1, we describe an algorithm that answers the question whether a given collection of bids can be represented in a matrix bid. If the answer is no, we need to find a matrix bid that offers a good approximation of the bids. This is what is done in section 6.4.2, by generalizing the algorithm of section 6.4.1.

6.4.1 Exact expression of a bid function using a matrix bid

From the description in section 6.1, it is clear that not every bid function can be represented by a matrix bid. This is due to the fact that there is in general only one entry available to express the added value of an item to many sets. Indeed, whereas there is only one set corresponding to the entries on the first column and to those on the diagonal, this is not the case for the other entries. Day (2004) shows that for each entry b_{ik}, the number of sets containing $k-1$ items ranked higher than i corresponds to Pascal's triangle[1], which implies that the entries towards the middle and the bottom depend on the largest number of sets (notice that we drop the index j again, since this section deals only with the bid function of a single bidder).

[1] see http://mathworld.wolfram.com/PascalsTriangle.html

As mentioned in section 6.1, an arbitrary bid function can be represented as a matrix bid if and only if the incremental value an item adds to a set is determined only by the number of higher ranked items in that set. Although a list of bidder types that can be represented in a matrix bid (see section 6.2) gives some insight in the expressiveness of the matrix bid language, a test to determine whether a given collection of bids can fit into a matrix bid would be useful as well. Indeed, in practice, bidders do not necessarily bid according to a well-known structure, but often simply express a number of bids on various sets of their interest.

Consider a given set of items G, with $|G| = m$, and bids $v(S)$ with $S \subseteq G$. A naive way of verifying whether the given bids can fit into a matrix bid involves checking every possible ranking of the items; this has a time complexity of $O(m!2^m)$. The following procedure FIT(p, G, v) tries to fill the first p rows of the matrix bid with items in G, such that the bids that follow from this matrix bid correspond with the given bids v. In other words, if such a matrix bid exists, FIT(m, G, v) returns YES and produces the ranking of the items and entries b_{ik}. Otherwise, FIT(m, G, v) returns NO, indicating that the given bids cannot be represented in a matrix bid.

Algorithm 1 FIT(p, G, v)

```
STEP 1:
$L \leftarrow G$; $A \leftarrow \{\}$;
Go to STEP 2;

STEP 2:
Pick an item $i$ from $L$;
$L \leftarrow L \setminus \{i\}$; $k \leftarrow 1$;
Go to STEP 3;

STEP 3:
$\exists b_{ik} \in \mathbb{R} : v(S \cup \{i\}) - v(S) = b_{ik}$ for all sets $S \subseteq G \setminus \{i\}$ with $|S| = k - 1$?
if YES then
  if $(k < p)$ then
    $k \leftarrow k + 1$;
    Go to STEP 3;
  else
    $A \leftarrow A \cup \{i\}$;
    if $(L \neq \{\})$ then
      Go to STEP 2;
    else
      Go to STEP 4;
    end if
  end if
end if
if NO then
  if $(L \neq \{\})$ then
    Go to STEP 2;
  else
    Go to STEP 4;
  end if
end if

STEP 4:
if $(A = \{\})$ then
  return NO;
else
  for $(i \in A)$ do
    $r_i = p$;
    Fill row $p$ with $b_{ik}$ with $1 \leqslant k \leqslant p$;
    $p \leftarrow p - 1$;
  end for
  if $(p = 0)$ then
    return YES;
  else
    FIT$(p, G \setminus A, v)$;
  end if
end if
```

Theorem 21. *FIT(m, G, v) recognizes whether a bid function b, defined on subsets of a set G, containing m items, can be expressed as a matrix bid.*

Proof. The algorithm FIT(p, G, v) searches for items in G that can be placed on the p-th row of the matrix bid. To keep track of these items, the algorithm uses the set A, which is initialized to the empty set in step 1. Furthermore, the algorithm initializes the set L of items that still need to be assigned to a row p. In step 2, an item is picked from L. In step 3, we verify whether a value can be found for an entry k on row p that satisfies the requirements that follow from the given bids v. By repeating step 3 for every entry of row p, we find out whether item i can be placed on row p; if this is the case, we add it to A. From the way a bid is computed from a matrix bid (see 6.1), it follows that the key condition is indeed:

$$\exists b_{ik} \in I\!R : v(S \cup \{i\}) - v(S) = b_{ik}, \forall S \subseteq G \setminus \{i\} : |S| = k - 1, \forall k \in \{1, ..., p\}?$$

This condition is checked for all items that have not yet been placed on a row of the matrix bid. If we find a non-empty set A in step 4, we can use the items in A to fill rows $p - |A| + 1$ to p, with the b_{ik} values that are obtained in step 3. Indeed, if an item can be placed on row p, it can also be placed on a higher ranked row (i.e. $p - 1$, $p - 2$, etc.) since the sets for which the condition in step 3 needs to be valid on such a row are a subset of the sets that need to satisfy the condition for row p. Notice that the choice of which item of A should be placed on which row from $p - |A| + 1$ to p has no effect on choices made regarding rows lower than p or higher than $p - |A| + 1$. At this point, we perform the FIT algorithm again, in order to place the remaining $G \setminus A$ items on the remaining $p - |A|$ rows. If we can continue the algorithm until all rows have been filled with an item, then clearly the given bids can be represented by a matrix bid.

If at some stage in step 4, the set A is empty, then we conclude that there is no way to represent the bids in a matrix bid. Indeed, this means that we found at some row q a set Q of items (with $|Q| = q$) that cannot be placed at that row q. We also know that none of these items can be placed at a row ranked lower than q (i.e. $q + 1$, $q + 2$, etc.), since otherwise this would have been clear at an earlier stage of the algorithm. Therefore, we can conclude that if the algorithm returns NO, no matrix bid exists that represents the given bids. □

In order to verify whether there exists an entry for item i on column k of row p, we need to check all sets of size $k - 1$ that do not contain item i. Thus, verifying whether an item fits on a row requires processing at most 2^m sets. As a result, this algorithm has a time complexity of $O(m^2 2^m)$, which is exponential in the number of items m, but polynomial in the input, being the number of bids, i.e. 2^m.

6.4.2 Approximate expression of a bid function using a matrix bid

In this section, we focus on the question how we can approximate a given collection of bids by a matrix bid. Day (2004) points out that such an approximation should ideally be safe and effective. An approximation is safe if it does not expose the bidder to paying more than he is willing to for any set. An approximation consisting of entries a_{ik} is effective if there exists no other safe approximation with entries b_{ik} such that $b_{ik} \geqslant a_{ik}$ for all $i \in G$ and all $k \in \{1, ..., r_i\}$ and $b_{ik} > a_{ik}$ for at least one $i \in G$ and $k \in \{1, ..., r_i\}$. Day (2004) also points out that the entries in the first column and on the diagonal can always be filled in exactly, since they correspond to only one set. However, no advice is given on how to choose one of the $m!$ possible rankings of the items.

The following algorithm APPROX(m, G, v) is an adaptation of the FIT algorithm, such that if no item is found that fits on some row of the matrix bid, an item q is selected and a safe approximation is constructed. Indeed, if no item fits on some row, there is for each item at least one entry on that row for which there is no single value that represents the marginal value of that item for all relevant sets. The selected item q is that item for which the sum of absolute values of the difference between the highest and the lowest marginal value is minimal. In case there are multiple such items, the algorithm makes an arbitrary choice among these items.

Algorithm 2 APPROX(m, G, v)

```
for ($p = m$ downto 1) do
  for ($i \in G$) do
    for ($k = 1$ to $p$) do
      $high(i,k) \leftarrow max_{S \subseteq G \setminus \{i\}:|S|=k-1}(v(S \cup \{i\}) - v(S))$;
      $low(i,k) \leftarrow min_{S \subseteq G \setminus \{i\}:|S|=k-1}(v(S \cup \{i\}) - v(S))$;
    end for
    $d(i) \leftarrow \sum_{k=1}^{p} |high(i,k) - low(i,k)|$;
  end for
  $Q \leftarrow \text{argmin}_i \, d(i)$;
  Pick an arbitrary $q \in Q$;
  $G \leftarrow G \setminus \{q\}$;
  $r_q = p$;
  for ($k = 1$ to $p$) do
    $b_{qk} \leftarrow low(q,k)$;
  end for
end for
```

Notice that the notion of a safe approximation is especially important when the matrix bid is expressed by the bidder and communicated to the auctioneer. However, it may also be the case that the bidder does not communicate his bids as a matrix bid, but that the auctioneer translates them to a matrix bid. The auctioneer might do this in order to benefit from the fact that the winner determination problem of a matrix bid auction can be solved more efficiently than that of a general auction (see Day & Raghavan (2006)). If this is the case, the approximation need not necessarily be safe, since the auctioneer could use it only to determine the allocation, and use the original bids to determine the actual prices. In this case, the choice for the average of the highest and the lowest marginal value (or even the highest marginal value) as an approximate entry b_{qk} in the matrix bid could be motivated as well.

6.5 Conclusion

In this chapter, we focussed on the matrix bid auction, which is a combinatorial auction where a restriction is imposed on the preferences that can be expressed by a bidder. We investigated the structure on the bids implied by this auction and revealed the relationship between a matrix bid and concepts like free disposal, complement freeness, decreasing marginal valuations, and gross substitutes. Finally, we developed tools that should facilitate the use of the matrix bid auction in practice. Given a number of bids, we found a way to determine whether these bids can be expressed as a matrix bid. Finally, we also show how to approximate these bids in a matrix bid if an exact representation is not possible.

Chapter 7

Exact algorithms for the matrix bid auction

In this chapter, we focus on solving the winner determination problem for the matrix bid auction (see chapter 6) exactly. In section 7.1, we discuss the computational complexity of the special case of the matrix bid auction winner determination problem that arises if all bidders have the same ranking of the items. Section 7.2 deals with two mathematical programming formulations for the general matrix bid auction winner determination problem. Based on one of these formulations, we develop two branch-and-price algorithms to solve the winner determination problem in section 7.3. Finally, in section 7.4, we present computational results for these algorithms and compare them with results from the branch-and-cut approach by Day (2004).

7.1 Computational complexity

In the matrix bid auction, the key assumption is that for each bidder, the extra value an item adds to a set depends only on the number of higher ranked items in that set, according to the ranking of that bidder. This assumption does not prevent the bidder from expressing a bid on any combination of items, but restricts what these bids can be. Despite this restriction, the winner determination problem of the matrix bid auction is NP-hard (Day 2004). Even if each bidder has the same ranking of the items, the matrix bid auction winner determination problem remains NP-hard. Moreover, unless $P = NP$, there exists no polynomial-time approximation scheme (PTAS) for this problem.

Theorem 22. *There exists no polynomial-time approximation scheme for the winner determination problem for the matrix bid auction where all bidders have an identical ranking of the items, unless $P = NP$.*

Proof. We consider the winner determination problem for the matrix bid auction where all bidders have an identical ranking. We refer to this problem as MBI. The reduction is from the 3-dimensional matching (3DM) problem. The 3DM problem is described as follows: given a set $M \subseteq X \times Y \times Z$ of triples, where each of the sets X, Y and Z has exactly q elements, find the largest matching in M. Kann (1991) shows that it is NP-hard to decide whether there exists a matching of size q, or whether every matching has a size of at most $(1 - \delta)q$ for some fixed $\delta > 0$ (see also Petrank (1994)).

Every instance of 3DM can be reduced to an MBI instance in polynomial time. Suppose that the $3q$ elements of the sets X, Y, and Z correspond to $3q$ items and that each 3-element subset in M corresponds to a bidder. We pick an arbitrary ordering of the items and let this be the ranking of the items for each bidder. Each bidder thus has a matrix bid with this ranking and with the following entries. The highest ranked item of the triple corresponding to the bidder gets a value of 1 in the first column, the second highest ranked item gets a value of 2 in the second column, and the third highest ranked item gets a value of 3 in the third column. All other entries get a value of zero.

If an instance of 3DM has a matching of size q, then the corresponding instance of MBI has a solution of value $6q$. Indeed, a solution of 3DM consists of q pairwise disjoint 3-element subsets, corresponding to q bidders in MBI. Each supplier has a bid of 6 for the 3 items represented by the 3-element subset. Accepting these bids leads to a sum of winning bids equal to $6q$. Since every element of $X \cup Y \cup Z$ occurs exactly once in the solution of 3DM, every item will also be auctioned exactly once in the MBI solution.

If our instance of 3DM has a matching of size at most $(1-\delta)q$, at most $(1-\delta)q$ entries with value 3 in the matrix bids can be used, resulting in a MBI solution value of $(1-\delta)6q$. Notice that for a maximal solution value, we need to use a maximal number of entries with value 3. The number of items remaining is $3q - 3(1-\delta)q = 3\delta q$. Each pair of these items adds at most 3 to the solution value, resulting in a maximal solution value for MBI of

$$(1-\delta)6q + \frac{9\delta q}{2} = (6 - \frac{3}{2}\delta)q.$$

Consequently, a polynomial-time approximation scheme for MBI would imply that we could distinguish between instances of 3DM with a matching of size q and instances where every matching has a size of at most $(1-\delta)q$, which is an NP-hard problem (Kann 1991). □

Notice that it follows from theorem 22 that the winner determination problem for the matrix bid auction where bidders have an identical ranking of the items is NP-hard. In this theorem, the number of bidders is part of the input. In the case that the number of bidders is fixed (and we still assume identical rankings), the winner determination problem can be solved in polynomial time.

Theorem 23. *The winner determination problem for a matrix bid auction with a fixed number of bidders, all having an identical ordering of the items,* *can be solved in polynomial time.*

Proof. We will show that the winner determination problem for a matrix bid auction with a fixed number of bidders, say n, all having an identical ranking r of the items, say $1, 2, ..., m$, can be solved by solving a longest path problem on an acyclic graph involving $O(m^{n+2})$ nodes and $O(nm^{n+2})$ arcs.

This graph contains nodes indexed by $(i, s_1, s_2, ..., s_n, k)$, a source, and a sink. The index i refers to item i and ranges from 1 to m. The indices s_j, with $j \in \{1, 2, ..., n\}$, and k range from 0 to r_i, with $\sum_j s_j + k = r_i$. There are arcs from each node $(i, s_1, s_2, ..., s_n, k)$ to $(i+1, s'_1, s'_2, ..., s'_n, k')$, provided that $s'_j \geqslant s_j$ for all $j \in \{1, 2, ..., n\}$, and that $k' \geqslant k$. Furthermore, there is an arc from the source to each node $(1, s_1, s_2, ..., s_n, k)$, and an arc from each node $(m, s_1, s_2, ..., s_n, k)$ to the sink. The arc from node $(i, s_1, s_2, ..., s_n, k)$ to $(i+1, s'_1, s'_2, ..., s'_n, k')$ has a cost of b_{i+1,j,s'_j} where j is the index for which $s'_j = s_j + 1$, if $k' = k$. If $k' = k + 1$, then this arc has a cost of zero. All arcs to the sink also have a cost of zero. The graph is depicted in Figure 7.1 for a setting with 2 items and 2 bidders. All arcs without indication of the corresponding cost have a cost equal to zero.

The graph described above should be interpreted as follows. Each node $(i, s_1, s_2, ..., s_n, k)$ corresponds to a state where a decision has been made on the allocation of item i and all items ranked higher than i, with each bidder j receiving s_j items and k items remaining with the auctioneer. Selecting an arc from $(i, s_1, s_2, ..., s_n, k)$ to $(i + 1, s'_1, s'_2, ..., s'_n, k')$ therefore corresponds to allocating item $i + 1$ to that bidder j for which $s'_j = s_j + 1$. If there is no such bidder, then item $i+1$ remains with the auctioneer (and $k' = k+1$). In this way, each path from source to sink determines how the items are to be allocated, and there is a path from the source to the sink for each possible allocation.

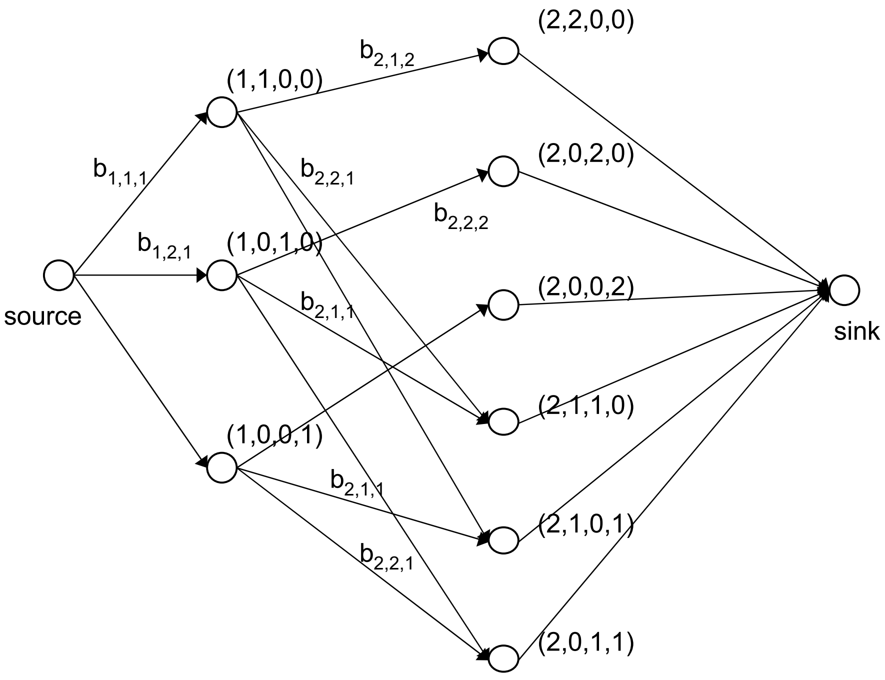

Figure 7.1: Illustration of the graph for 2 items and 2 bidders

We now sketch the equivalence between the length of a path in the graph and the value of an allocation of the items. We know that in a matrix bid, the value of adding an item i to a set is determined only by the number of higher ranked items. Since the graph contains only arcs from higher ranked items to lower ranked items, the effect of adding an item i to a set on the bid for this set can be determined, regardless of whatever items are added to the set further down the path. The cost of an arc is nothing else but the appropriate entry from the matrix bid of the bidder receiving the item. This means that the length of any path from source to sink corresponds to $\sum_j b_j(S_j)$, where S_j is the set of items allocated to bidder j, according to that path. Therefore, the winner determination problem for a matrix bid auction with a fixed number of bidders, all having an identical ordering of the items, can be solved by solving a longest path problem. This can be done in polynomial time, since the underlying graph is acyclic. □

7.2 Mathematical formulations

In this section, we present two mathematical formulations for the matrix bid auction winner determination problem. The first formulation (see also Day (2004)) is inspired by the assignment problem, the second by the set packing problem. We show that the LP-relaxations of both formulations are equally strong.

We define the binary variable x_{ijk} to be 1 if bidder j receives item i as the k-th best item, and 0 otherwise. This leads to the formulation below, to which we refer as the assignment formulation.

maximize

$$\sum_{i \in G} \sum_{j \in B} \sum_{k=1}^{r_{ij}} b_{ijk} x_{ijk} \tag{7.1}$$

subject to

$$\sum_{j \in B} \sum_{k=1}^{r_{ij}} x_{ijk} \leqslant 1 \qquad \forall i \in G \tag{7.2}$$

$$\sum_{i \in G : r_{ij} \geq k} x_{ijk} \leqslant 1 \qquad \forall j \in B, \forall k \in \{1, ..., r_{ij}\} \tag{7.3}$$

$$\sum_{l \in G : k \leq r_{lj} \leq r_{ij}} x_{ljk} \leqslant \sum_{l \in G : k-1 \leq r_{lj} < r_{ij}} x_{ljk-1} \qquad \forall i \in G, \forall j \in B, \forall k \in \{2, ..., r_{ij}\} \tag{7.4}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i \in G, \forall j \in B, \forall k \in \{1, ..., r_{ij}\} \tag{7.5}$$

Constraints (7.2) enforce that each item can be assigned to at most one bidder, while constraints (7.3) make sure that for each bidder, at most one item is the k-th best item in the set this bidder gets. Finally, constraints (7.4) impose that a bidder cannot get an item as the k-th best item in a set, unless a higher ranked item was assigned to this bidder as his $(k-1)$-th best item in this set. Constraints (7.5) are the integrality constraints.

Notice that the formulation (7.1)-(7.5) is not the minimal correct formulation for the matrix bid winner determination problem. Indeed, constraints (7.3) for $k \in \{2, ..., r_{ij}\}$ are redundant in (7.1)-(7.5), since they are already enforced by constraints (7.3) for $k = 1$ and constraints (7.4). Also, replacing constraints (7.4) with the following (weaker) constraints still results in a correct formulation:

$$x_{ijk} \leqslant \sum_{l \in G: k-1 \leq r_{lj} < r_{ij}} x_{ljk-1} \qquad \forall i \in G, \forall j \in B, \forall k \in \{2, ..., r_{ij}\}.$$

However, with this formulation, all constraints (7.3) remain necessary.

The set packing formulation below makes use of binary variables $y(S, j)$, which equals 1 if bidder j wins set S, and 0 otherwise. The first set of constraints (7.7) enforces that each item is awarded to at most one bidder. The second set of constraints guarantees (7.8) that no bidder receives more than one set. The integrality constraints are (7.9).

maximize

$$\sum_{j \in B} \sum_{S \subseteq G} b_j(S) y(S, j) \tag{7.6}$$

subject to

$$\sum_{S \supseteq \{i\}} \sum_{j \in B} y(S, j) \leqslant 1 \qquad \forall i \in G \tag{7.7}$$

$$\sum_{S \subseteq G} y(S, j) \leqslant 1 \qquad \forall j \in B \tag{7.8}$$

$$y(S, j) \in \{0, 1\} \qquad \forall S \subseteq G, \forall j \in B \tag{7.9}$$

Notice that this set packing formulation is identical to the formulation (1.1)-(1.4) for the winner determination problem of a general combinatorial auction, with $\Omega_j = 2^G$. Indeed, the matrix bid auction only differs from a general combinatorial auction in the way $b_j(S)$ is computed. Notice also that the assignment formulation is polynomially sized in the number of bidders and the number of items. This is not the case for the set packing formulation. In the following theorem, we prove that the LP-relaxation of the set packing formulation and the LP-relaxation of the assignment formulation are equally strong.

Theorem 24. *The LP relaxation of the assignment formulation and the LP relaxation of the set packing formulation are equally strong. Moreover, if the assignment formulation has an integral solution that is optimal with respect to the LP-relaxation, this is also the case for the assignment formulation, and vice versa.*

Proof. In order to prove the first part of this theorem, we need to show that the LP-relaxation of the set packing formulation is at least as strong as the LP-relaxation of the assignment formulation and vice versa. In order to prove the first relation, we need to show that any solution $\hat{y}$ of the LP-relaxation of the set packing formulation can be transformed to a solution $\hat{x}$ of the LP-relaxation of the assignment formulation with the same objective function value. This is accomplished by the following procedure. For the remainder of this proof, if we mention a formulation, we mean in fact its LP-relaxation.

First, we initialize all variables $\hat{x}_{ijk}$ to 0, for all $i \in G$, $j \in B$, and $k \in \{1, ..., r_{ij}\}$. We consider each variable $\hat{y}(S, j)$, with $S \subseteq G$ and $j \in B$ once, and set for each item i in S

$$\hat{x}_{i,j,k(i,j,S)} \leftarrow \hat{x}_{i,j,k(i,j,S)} + \hat{y}(S,j). \tag{7.10}$$

Thus, in this procedure, the value of each variable $\hat{y}(S,j)$ is added to $|S|$ $\hat{x}_{ijk}$ variables, namely those with item $i \in S$, and $k = k(i,j,S)$. It follows that the following equality is valid:

$$\sum_{k=1}^{r_{ij}} \hat{x}_{ijk} = \sum_{k=1}^{r_{ij}} \sum_{S:i\in S \wedge k=k(i,j,S)} \hat{y}(S,j) = \sum_{S\supseteq\{i\}} \hat{y}(S,j) \qquad \forall i \in G, j \in B. \tag{7.11}$$

Using this equality, we verify that (7.2) holds for $\hat{x}$:

$$\sum_{j\in B}\sum_{k=1}^{r_{ij}} \hat{x}_{ijk} = \sum_{j\in B}\sum_{S\supseteq\{i\}} \hat{y}(S,j) \leqslant 1. \tag{7.12}$$

Notice that the last inequality follows from the feasibility of $\hat{y}$ (see (7.7)). We also establish for $j \in B$, and $k \in \{1,...,r_{ij}\}$:

$$\begin{aligned} \sum_{i\in G:r_{ij}\geqslant k} \hat{x}_{ijk} &= \sum_{i\in G}\sum_{S:i\in S\wedge k(i,j,S)=k} \hat{y}(S,j) \\ &= \sum_{S:|S|\geqslant k} \hat{y}(S,j) \\ &\leqslant \sum_{S\subseteq G} \hat{y}(S,j) \leqslant 1, \end{aligned} \tag{7.13}$$

which shows that $\hat{x}$ satisfies (7.3). Finally, we have that for each $i \in G$, $j \in B$, and $k = 1,...,r_{ij}$:

$$\sum_{l\in G:k\leq r_{lj}\leq r_{ij}} \hat{x}_{ljk} = \sum_{l\in G:r_{lj}\leqslant r_{ij}} \sum_{S:l\in S\wedge k(l,j,S)=k} \hat{y}(S,j). \tag{7.14}$$

Thus we can write for each $i \in G$, $j \in B$, and $k = 2,...,r_{ij}$:

$$\sum_{l\in G:k\leq r_{lj}\leq r_{ij}} \hat{x}_{ljk} - \sum_{l\in G:k-1\leq r_{lj}<r_{ij}} \hat{x}_{ljk-1} =$$
$$\sum_{l\in G:r_{lj}\leqslant r_{ij}} \sum_{S:l\in S\wedge k(l,j,S)=k} \hat{y}(S,j) - \sum_{l\in G:r_{lj}\leqslant r_{ij}} \sum_{S:l\in S\wedge k(l,j,S)=k-1} \hat{y}(S,j). \tag{7.15}$$

Consider some $\hat{y}(S,j)$ occurring in the first term. The corresponding set S has at the k-th position ($k \geqslant 2$) some item l, $r_{lj} \leqslant r_{ij}$. It follows that there must be some other item, say l' with $r_{l'j} \leqslant r_{lj}$ at position $k-1$. Hence this $\hat{y}(S,j)$ also occurs in the second term. It follows that the expression (7.15) cannot have a positive value, and hence (7.4) is satisfied. Notice also that the transformation procedure (7.10) does not affect the objective function value. Moreover, it transforms any integral solution $\hat{y}$ to an integral solution $\hat{x}$.

Hence, we have shown that the set packing formulation is at least as strong as the assignment formulation and if the set packing formulation has an integral solution that is optimal with respect to the LP-relaxation, this is also the case for the assignment formulation. In the remainder of this proof, we show that the assignment formulation is at least strong as the set packing formulation. In order to prove this second relation, we show that any solution $\hat{x}$ of the LP-relaxation of the assignment formulation can be transformed to a solution $\hat{y}$ of the LP-relaxation of the set packing formulation with the same objective function value. This is accomplished by the following procedure, CONVERT($\hat{x}$).

Algorithm 3 CONVERT($\hat{x}$)

```
for ($j \in B$) do
  Initialize $\hat{y}(S, j) \leftarrow 0$ for all $S \subseteq G$;

  Step 1:
  for ($i \in G$) do
    $\hat{y}(\{i\}, j) \leftarrow \hat{x}_{ij1}$
  end for

  Step 2:
  for ($k = 2$ to $m$) do
    for ($i \in G$: $r_{ij} = k$ to $m$) do
      Step 2a:
      $T = \{S \subseteq \{i' : r_{i'j} < r_{ij}\} : |S| = k - 1\}$;
      while ($\hat{x}_{ijk} > 0$) do
        Pick a set $S$ from $T$ and remove $S$ from $T$;
        if ($\hat{x}_{ijk} > \hat{y}(S, j)$) then
          $\hat{y}(S \cup \{i\}, j) \leftarrow \hat{y}(S, j)$;
          $\hat{x}_{ijk} \leftarrow \hat{x}_{ijk} - \hat{y}(S, j)$;
          $\hat{y}(S, j) \leftarrow 0$;
        else
          $\hat{y}(S \cup \{i\}, j) \leftarrow \hat{x}_{ijk}$;
          $\hat{y}(S, j) \leftarrow \hat{y}(S, j) - \hat{x}_{ijk}$;
          $\hat{x}_{ijk} \leftarrow 0$;
        end if
      end while
    end for
  end for
end for
```

The CONVERT procedure translates any solution for the assignment formulation to a solution for set packing formulation. First, we argue that the CONVERT algorithm terminates.

The crucial step in the CONVERT algorithm is step 2a, which has to be performed for each bidder j, for each k from 2 to m, and for each $i \in G$ with $r_{ij} \geqslant k$. Let us consider now a bidder j, item i, and rank k, for which step 2a is to be performed, and let $\widetilde{y}(S,j)$ be the solution as it is constructed by the CONVERT algorithm so far. In order to guarantee that the while loop in step 2a terminates, we need:

$$\hat{x}_{ijk} \leqslant \sum_{S:S \subseteq \{i':r_{i'j}<r_{ij}\} \wedge |S|=k-1} \widetilde{y}(S,j). \tag{7.16}$$

Notice that in CONVERT, so far, each variable $\widetilde{y}(S,j)$, with $|S| = k-1$ and l being the lowest ranked item in S, has been increased at most once, namely with (a fraction of) $\hat{x}_{l,j,k-1}$. Furthermore, the total value of $\hat{x}_{l,j,k-1}$ has been added exclusively over variables $\widetilde{y}(S,j)$ with $|S| = k-1$ and l the lowest ranked item in S. Therefore, we have that the total fraction that has been added to variables $\widetilde{y}(S,j)$ with S containing $k-1$ items ranked higher than i equals:

$$\sum_{i':k \leqslant r_{i'j} < r_{ij}} \hat{x}_{i',j,k-1}. \tag{7.17}$$

Notice that the value of each variable $\widetilde{y}(S,j)$ may also have been decreased in CONVERT. Indeed, variables $\widetilde{y}(S,j)$ with S containing $k-1$ items and the one with the lowest rank being l, can be decreased only with (a fraction of) variables $\hat{x}_{i',j,k}$ with i' ranked higher than l, and lower than i (since step 2a has not yet been performed for rank k and item i or items ranked lower than i). Furthermore, the total value of $\hat{x}_{l,j,k}$ has been subtracted only from variables $\widetilde{y}(S,j)$, with S containing $k-1$ items, all ranked higher than l. Therefore, we have that the total fraction that has been subtracted from variables $\widetilde{y}(S,j)$ with S containing $k-1$ items ranked higher than i equals:

$$\sum_{i':k \leqslant r_{i'j} < r_{ij}} \hat{x}_{i',j,k}. \tag{7.18}$$

Thus,

$$\sum_{S: S \subseteq \{i': r_{i'j} < r_{ij}\} \wedge |S| = k-1} \widetilde{y}(S,j) = \sum_{i':k \leqslant r_{i'j} < r_{ij}} \hat{x}_{i',j,k-1} - \sum_{i':k \leqslant r_{i'j} < r_{ij}} \hat{x}_{i',j,k}. \tag{7.19}$$

Further, it follows from (7.4) that

$$\hat{x}_{ijk} \leqslant \sum_{i':k \leqslant r_{i'j} < r_{ij}} \hat{x}_{i',j,k-1} - \sum_{i':k \leqslant r_{i'j} < r_{ij}} \hat{x}_{i',j,k} \tag{7.20}$$

for each bidder j, for each k from 2 to m, and for each $i \in G$ with $r_{ij} \geqslant k$. From (7.19) and (7.20) we conclude that (7.16) is true and hence the CONVERT algorithm terminates.

We now argue that solution $\hat{y}$ is indeed feasible with respect to constraints (7.7), (7.8), and the relaxation of (7.9).

For each bidder j and each item i, it is clear that after step 1, $\sum_{S \supseteq \{i\}} \hat{y}(S,j) = \hat{x}_{ij1}$. In step 2, each value $\hat{x}_{ijk}$ is spread over one or more variables $\hat{y}(S,j)$ with S containing item i. Also, for each variable $\hat{y}(S,j)$ that is increased, a variable $\hat{y}(S \setminus \{i\}, j)$ is decreased with the same value. Therefore, after step 2, $\sum_{S \supseteq \{i\}} \hat{y}(S,j) \leqslant \sum_{k=1}^{r_{ij}} \hat{x}_{ijk}$. Summing over the bidders gives $\sum_{j \in B} \sum_{S \supseteq \{i\}} \hat{y}(S,j) \leqslant \sum_{j \in B} \sum_{k=1}^{r_{ij}} \hat{x}_{ijk}$. Given (7.2), this implies that constraints (7.7) are satisfied.

For each bidder j, it is clear that after step 1 of CONVERT, $\sum_{S \subseteq G} \hat{y}(S,j) = \sum_{i \in G} \hat{x}_{ij1}$. In step 2, for every variable $\hat{y}(S,j)$ whose value is increased, there is some other variable $\hat{y}(S',j)$ whose value is reduced by the same amount. Given (7.3), this implies that constraints (7.8) are satisfied.

Each variable $\hat{y}(S,j)$ is increased by at most one variable $\hat{x}_{ijk}$. Therefore, it follows from the relaxation of constraints (7.5) that $\hat{y}(S,j) \leqslant 1$ for all $S \subseteq G$ and each bidder j. By construction of the algorithm, no variable $\hat{y}(S,j)$ will have a value less than zero. Thus, $\hat{y}$ satisfies the relaxation of constraints (7.9).

Further, the objective function value of both solutions $\hat{x}$ and $\hat{y}$ is the same. Consider any bidder j. After step 1, the objective function of solution $\hat{y}$ has a value equal to $\sum_{i \in G} b_{ij1}\hat{x}_{ij1}$, since $b(\{i\},j) = b_{ij1}$. Every time step 2a is performed, the objective function value is increased by $(b(S \cup \{i\},j) - b(S,j))\hat{x}_{ijk}$. Since set S contains only items ranked higher than item i, we have $b(S \cup \{i\},j) - b(S,j) = b_{ijk}$, where k is the number of items in S plus one. Therefore, after step 2 the objective function equals $\sum_{i \in G} \sum_{k=1}^{r_{ij}} b_{ij1}\hat{x}_{ij1}$. Summing over all bidders j shows that the CONVERT($\hat{x}$) procedure produces a solution $\hat{y}$ with the same objective function value as $\hat{x}$.

Finally, it is easy to see that if the CONVERT procedure is confronted with an integral solution $\hat{x}$, it will produce an integral solution $\hat{y}$. Thus, we can conclude that the assignment formulation and the set packing formulation are equally strong, and that if one formulation has an integral optimal solution, this is also the case for the other formulation. □

7.3 Branch-and-price algorithms for solving the matrix bid auction

Theorem 24 shows that the set packing formulation (7.6)-(7.9) is equally strong as the assignment formulation (7.1)-(7.5). Here we outline an algorithm based on the set packing formulation. Solving the LP-relaxation of the set packing formulation is however not trivial, given the huge amount of variables ($n2^m$). Considering that only a small percentage of these variables are nonzero in an optimal solution, column generation suggests itself as an efficient solution technique. Column generation was proposed by Dantzig & Wolfe (1960) and starts by solving the LP-relaxation considering only a restricted subset of the variables. This problem is also called the restricted master problem. Notice that this problem can be restricted to $m + n$ variables, whereas the assignment formulation requires $nm(m + 1)/2$ variables, which may still be large. The next step is to verify whether any of the variables that were not considered could improve the current solution. In other words, we search for a variable with a non-negative reduced cost. This problem is called the pricing problem. If we find such a variable, we add it to the restricted master problem and solve it again. This re-optimizing and pricing is to be repeated until the pricing problem fails to produce new variables, indicating that the LP-relaxation has been solved to optimality.

Notice that the column generation procedure does not guarantee to find an integral solution. In case of a fractional solution, a branching decision needs to be made, partitioning the solution space in order to create a number of smaller subproblems. With branch-and-price, this results in a search tree where column generation has to be applied in every node. In this way, branch-and-price can be seen as a generalization of the column generation technique for integer programming. Combining the column generation approach with a branching scheme may not be straightforward. The key to an efficient branch-and-price algorithm is an easy-to-solve pricing problem. The branching rule should therefore not destroy the structure of the pricing problem or increase its complexity when moving deeper down the search tree.

Branch-and-price has proven to be successful for solving huge integer programs arising from a number of combinatorial problems (see Barnhart, Johnson, Nemhauser, Savelsbergh & Vance (1998) for an overview). We refer to Vanderbeck & Wolsey (1996) for a more elaborate description of the branch-and-price technique. In section 7.3.1, we show how the LP-relaxation of the set packing formulation for the matrix bid auction winner determination problem can be solved efficiently using column generation. Next, the column generation approach is used as a building block for two branch-and-price algorithms to solve the matrix bid auction. The algorithm in section 7.3.2 makes use of a branching rule based on assigning items to bidders, whereas in section 7.3.3, branching is done by deciding on the succession of items in a winning set. Finally, in section 7.3.4, we comment on some issues that turn out to be important while implementing both branch-and-price algorithms.

7.3.1 Column generation for the matrix bid auction

In this section, we show how the LP-relaxation of the set packing formulation of the matrix bid winner determination problem can be solved using column generation. We also prove that the pricing problem can be solved in polynomial time, since it can be solved by solving a shortest path problem.

If we define u_i for each item $i \in G$ as the dual price associated with the corresponding constraint of (7.7), and v_j for each bidder $j \in B$ as the dual price associated with the corresponding constraint of (7.8), we can write the dual of the set packing formulation (7.6)-(7.9) as follows:

minimize

$$\sum_{i \in G} u_i + \sum_{j \in B} v_j \tag{7.21}$$

subject to

$$\sum_{i \in S} u_i + v_j \geqslant b_j(S) \qquad \forall S \subseteq G, \forall j \in B \tag{7.22}$$

$$u_i \geqslant 0, v_j \geqslant 0 \qquad \forall i \in G, \forall j \in B \tag{7.23}$$

We start by finding an optimal solution for the restricted master problem, i.e. the LP-relaxation of (7.6)-(7.9) considering only a limited number of variables $y(S,j)$. This solution is also an optimal solution for the (unrestricted) LP-relaxation of (7.6)-(7.9) if its corresponding dual variables form a feasible solution for (7.21)-(7.23), which has a constraint for every variable $y(S,j)$. Consequently, we need to add a new column or variable to the restricted master problem if a constraint of (7.22) is violated. The pricing problem thus boils down to determining the existence of a set S of items and a bidder j such that

$$\sum_{i \in S} u_i < b_j(S) - v_j. \tag{7.24}$$

Theorem 25. *The pricing problem, i.e. finding a set S of items and a bidder j such that a constraint of (7.22) is violated, can be solved by solving a shortest path problem.*

Proof. We construct a graph with a source and a sink, and a subgraph for each bidder j. Such a subgraph contains r_{ij} nodes for each item i, called item nodes. We will refer to an item node as (i,j,k), where i stands for the item and k ranges from 1 to r_{ij}. There are arcs from each node (i,j,k) to each node $(i',j,k+1)$ where item i' is ranked lower than item i (i.e., $r_{i'j} > r_{ij}$). These arcs have a cost equal to $u_{i'} - b_{i',j,k+1}$. Notice that there are no arcs between nodes corresponding to different subgraphs. Furthermore, for each subgraph, there are arcs from the source node to node $(i,j,1)$ for each item i with a cost equal to $u_i - b_{ij1}$ and there are arcs from each item node (i,j,k) to the sink with cost v_j. A schematic representation of this graph is given in Figure 7.2 for a setting with a single bidder j and three items.

From the structure of this graph, it follows that all nodes of a path from the source to the sink correspond to the same bidder and each path contains at most one node per item. Moreover, exactly one arc with cost v_j is included

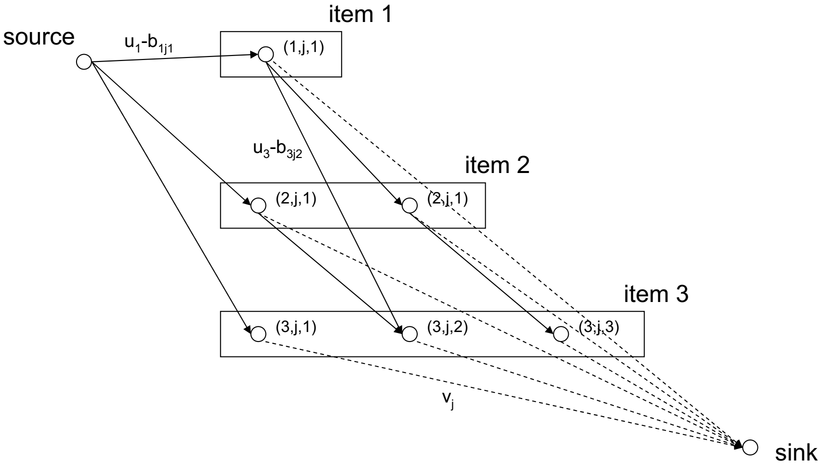

Figure 7.2: The pricing problem as a shortest path problem

in the path. Therefore, the length of a path containing nodes (i, j, k) of the items $i \in S$ of bidder j in this graph equals

$$\sum_{i \in S} (u_i - b_{ijk}) + v_j \tag{7.25}$$

Furthermore, the graph ensures that an item i is in the path using its k-th node only if a higher ranked item is in the path through its $(k-1)$-th node. We can therefore state that $\sum_{i \in S} b_{ijk} = b_j(S)$ and it follows that the existence of a path with negative length corresponds to a violated constraint in the dual. Consequently, we need to solve a shortest path problem on an acyclic graph in order to solve the pricing problem. □

Thus, if the shortest path has a negative length, we can add a column for the corresponding bidder j containing the items in set S determined by the item nodes traversed in the path. Naturally, bidder j's bid for this set S is

$b_j(S)$. Notice that since the pricing problem is solvable in polynomial time, the LP-relaxation of the set packing formulation for the matrix bid auction can also be solved in polynomial time.

Corollary 3. *The LP-relaxation of the set packing formulation (7.6)-(7.9) for the matrix bid auction winner determination problem can be solved in polynomial time.*

7.3.2 Branching on an item-bidder pair

The solution of the LP-relaxation of the matrix bid winner determination problem found by column generation may not be integral. If this is the case, we need to partition the solution space to eliminate this fractional solution. In this approach, we partition the solution space by the branching decision whether or not to assign an item to a bidder. We first prove that in a fractional solution, there always exists an item that has been fractionally assigned to one or more bidders.

Lemma 1. *For any fractional solution to the relaxation of (7.6)-(7.9),*

$$\exists i \in G, j \in B : 0 < \sum_{S:S \supseteq \{i\}} y(S,j) < 1 \tag{7.26}$$

Proof. We will prove this theorem by showing that a solution must be integral if it does not satisfy (7.26). Consider a solution for which property (7.26) is not valid. This means that each item has been assigned fully or not at all to each bidder. In this case, no items are split over multiple bidders. An item p for which $\sum_{S:S \supseteq \{p\}} y(S,j) = 1$ could, however, still be split over multiple sets of the same bidder j. It is easy to see that if bidder j is awarded a set S containing next to p any other item q, that this item then should occur in each set containing p in order to have the sum of the fractions of sets containing p equal 1. In other words, the sets of bidder j are identical, and we have, in fact, an integral solution. □

The branch-and-price algorithm can, however, only be valid if in every node of the search tree, all generated columns satisfy the previously made branching decisions. Prohibiting that an item is awarded to a certain bidder in the pricing problem can be done by simply removing the vertices corresponding to that item for that bidder from the graph. Enforcing that an item is awarded to a certain bidder in the pricing problem is less obvious. For that bidder, the arcs from the source to any lower ranked item need to be removed. Also the arcs from any higher ranked item to any item ranked lower than that item need to be deleted. Finally, the arcs from the higher ranked items to the sink must be removed as well. Clearly, all nodes that can no longer be reached as a consequence of these removals can now also be deleted, as are the arcs leaving those nodes, and so on. For all other bidders, we need to remove the vertices of that item from the graph. Figure 7.3 shows the pricing problem where item 2 is forced to be awarded to the bidder whose item nodes are depicted. In this graph, we made sure that every path from the source to the sink of that bidder must include a node corresponding to item 2.

Notice that this branching rule does not destroy the structure of the pricing problem: in all branches, the pricing problem remains a shortest path problem. It is easy to see that this shortest path problem can be adjusted to produce columns that comply with a series of branching decisions. Moreover, when moving deeper down the tree, more and more arcs and nodes will be removed. Thus, we have described a valid branching rule where the pricing problem remains solvable as a shortest path problem throughout the search tree.

7.3.3 Branching on a pair of successive items

Ryan & Foster (1981) suggest a branching rule for the set partitioning problems where two constraints are covered together or not at all by the variables

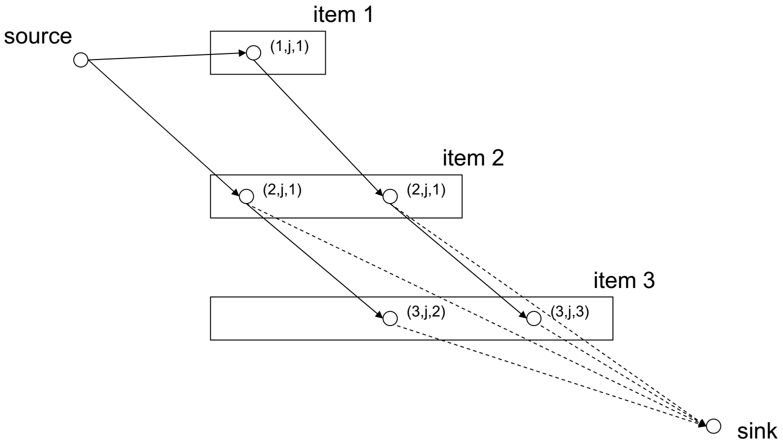

Figure 7.3: The pricing problem where the bidder must get item 2

in one branch, whereas in the other branch, each variable can cover at most one of these constraints. This rule can easily be generalized to set packing problems and can be translated to a combinatorial auction context as two items needing to go to the same bidder in one branch and to different bidders in the other branch. However, forcing two arbitrary items to go to the same bidder, but also forbidding that these items go to the same bidder, is not straightforward to achieve in the shortest path problem described in section 7.3.1. Therefore, we modify this branching rule, such that it takes into account the ranking of the items specified in the bidder's matrix bid. A similar modification has been applied for example in a pallet loading application (Moonen 2005).

We partition the solution space by branching on a pair of items p and q. In one branch, we enforce that if item p is present in a bidder's set, then item q must be directly successive to p in this set, when the set is sorted according to this bidder's ranking of the items. In the other branch, no bidder can have items p and q as direct successors in a set, according to his ranking.

We first prove that there always exists a pair of items such that the sets in which these items occur as direct successors according to the corresponding bidder's ranking, have been fractionally assigned to one or more bidders. We introduce the notation $p \rightarrow_j q$ to denote that item p is directly succeeded by item q in a set, according to the ranking of bidder j.

Lemma 2. *For any optimal, extreme fractional solution to the relaxation of (7.6)-(7.9),*

$$\exists p, q \in G : 0 < \sum_{j \in B} \sum_{S : S \supseteq \{p,q\} \wedge p \rightarrow_j q} y(S, j) < 1 \qquad (7.27)$$

Proof. Assume that we have an optimal, extreme fractional solution for which (7.27) is not satisfied. This means that for each pair of items, each bid on a set in which these items are direct successors according to ranking of the bidder that made the bid, has been assigned to that bidder for a total fraction of 0 or 1. Thus, for any items p and q for which $\sum_{j \in B} \sum_{S : S \supseteq \{p,q\} \wedge p \rightarrow_j q} y(S, j) = 1$, we can conclude that if item p is present in a set, that then also item q is present in this set. Therefore, each pair of sets to which a positive fraction has been assigned is disjoint or identical. Since there is a single variable $y(S, j)$ representing identical sets of the same bidder j, we conclude that identical sets must be split over multiple bidders. This leaves us with the problem of assigning a number of disjoint sets among one or more bidders, where each assignment of a set to a bidder has its profit, namely the bid of this bidder for this set. This problem is a maximum weighted assignment problem on a bipartite graph, where each node on one side of the partition represents a set, and each node on the other side of the partition represents a bidder. It follows that each optimal, extreme solution is integral. Consequently, for any optimal, extreme fractional solution to the relaxation of (7.6)-(7.9), property (7.27) is true. □

The above theorem shows that it is always possible to find a pair of items p and q on which to branch. However, we still need to enforce that the pricing problem will generate columns that satisfy the constraint imposed by the branching decision. In the branch where we impose $p \rightarrow_j q$, for each bidder j, we need to remove all arcs from nodes corresponding to p to any node not corresponding to q. Notice that for a bidder that ranks q higher than p, this comes down to removing all nodes related to p from the graph. This leaves us with a graph where if one arrives in a node related to p, the only option is to take an arc to a node related to q. In the branch where p should not be directly succeeded by q, it suffices, for each bidder, to remove the arcs going from a p-node to a q-node, if they exist.

Notice this branching rule does not destroy the structure of the pricing problem either, even when we consider a sequence of branching decisions. Indeed, it is not hard to verify that when going deeper into the search tree, the pricing problem can still be solved as a shortest path problem on an increasingly smaller graph.

7.3.4 Implementation issues

Both branch-and-price algorithms were implemented using Visual C++ 6.0. The set packing problems were solved using Ilog Cplex 8.1. The LEDA libraries (version 5.0.1) allowed us to solve the shortest path problems in linear time. In the remainder of this section, some of the most important implementation issues are discussed.

Solving the root node

A first issue that needs to be solved is determining which columns will be used in the very first restricted master problem. Using many columns obviously increases the computation time needed to solve the restricted master problem. On the other hand, this may result in a solution that is closer to the optimal solution, such that less iterations for solving the pricing problem and re-optimizing are needed. In our case, after experimenting with a number of settings, it turned out that including a rather large number of variables to start the column generation process pays off. We constructed a set for every strictly positive entry in the matrix bid by taking the item corresponding to this entry and completing the set with the k highest ranked items, where k is the entry's column in the matrix bid.

After the restricted master problem has been solved and the corresponding dual solution has been obtained, new columns with a non-negative reduced cost need to be added. The question remains how many such columns we should add. Again, adding too many new variables increases the computation time for solving the resulting restricted master problem, whereas adding too few variables can result in a large number of iterations for solving the pricing problem and re-optimizing. The strategy that proved to be the most efficient consists of adding for each bidder those variables whose reduced cost is at most 2% less than the most positive reduced cost for a variable from that bidder. Furthermore, the number of such variables that is added for each bidder cannot exceed the number of items. Notice that finding these variables demands very little extra computation time, since the LEDA libraries provide the distance from the source to each node in the graph, after having solved the shortest path problem.

Finally, when re-optimizing the restricted master problem, we start from the optimal base of the previous iteration. In order not to drag along too many columns for the remainder of the search tree, those columns that were added at some iteration, but never made part of any base solution are removed from the model. We keep the other columns, assuming that they will be useful again later.

A selection rule when branching on an item-bidder pair

The major issue in implementing this branching rule is to choose the item on which to branch and the bidder(s) to assign it to. We chose to branch on the item that is fractionally assigned to the highest number of bidders. For each of these bidders, a branch is constructed in which the bidder is assigned the item. A final branch is added where none of these bidders is allowed to receive the item. We opted for a depth-first strategy, where the branch where the item is assigned to the bidder with the highest fraction is explored first. Thus, the branch where bidders are disallowed to receive an item always comes last.

A selection rule when branching on a pair of successive items

With this branching rule, each node that needs further partitioning of the solution space leads to two branches. In the first branch, we enforce that for each bidder, if item p is present in a bid, q should be the next item in that bid, according to the ranking of that bidder. The second branch considers only bids for which p and q are no direct successors according to the bidder's ranking. We again chose a depth-first strategy, where the branch where $p \rightarrow_j q$ is imposed is explored first. The question remains how to select the items p and q. We opted to pick those items p and q for which $\sum_{j \in B} \sum_{S: S \supseteq \{p,q\} \wedge p \rightarrow_j q} y(S,j)$ is closest to 0.5.

Solving a tree node

Before we can start solving a node of the tree, we remove all columns that do not satisfy the latest branching decision. In case of backtracking, this branching decision expires and those columns are re-entered into the model, since we experienced that they often turn out to be useful in other branches of the tree.

The LP objective value of the node can be used as an upper bound to the integral solution that could be found further down the tree. Clearly, if this value is lower than the incumbent found so far, the node can be pruned. It may, however, require a large number of iterations to prove LP optimality. Vanderbeck & Wolsey (1996) show that the Lagrangian relaxation can also be used as an upper bound. The Lagrangian upper bound can be computed as (see e.g. Beliën (2006))

$$\delta + \sum_{j \in B} \max_{S \subseteq G} (RC(S,j), 0) \tag{7.28}$$

where δ is the objective value of the restricted master and $RC(S,j)$ is the reduced cost of variable $y(S,j)$. Notice that the computation of this bound requires little additional computational effort, since the pricing problem, which is solved for every bidder j anyway, finds the variable with the highest reduced cost. This upper bound is referred to as the Lagrangian upper bound, since it equals the bound obtained by Lagrange relaxation (Lasdon 1970). If at any iteration in the column generation process, the Lagrangian upper bound is lower than the incumbent, we can prune the node, without any risk of missing the optimal solution.

Obviously, when we re-optimize the restricted master problem, we also start from the optimal base of the previous iteration. The first restricted master problem is solved starting from the base solution of the parent node. Furthermore, as in the root node, we delete the added columns that turned out not to be useful.

7.4 Computational results

In this section, we elaborate on how we generated the instances on which the branch-and-price algorithms were tested. We also give an overview of the computational results and compare them with results from a branch-and-cut approach performed on the assignment formulation.

7.4.1 Structure of the instances

Unfortunately, real-life data for combinatorial auctions are not abundantly available for the public. It is therefore not uncommon in combinatorial auction literature to turn to randomly generated data (see for instance Leyton-Brown et al. (2000), Sandholm (2002), and Parkes (1999)). For a thorough discussion on the empirical hardness of several data distributions commonly used for combinatorial auctions, we refer to Leyton-Brown, Nudelman & Shoham (2005).

The randomly generated data we use, are due to Day (2004). Each matrix bid is composed according to a bid type, randomly chosen out of the six possibilities discussed in section 6.2 (additive preference bids, single-minded bids, nested flat bids, nested k-of bids, partition bids, and add-on bids) and a bid type that has non-increasing rows and columns. In order to avoid auctions for which the exact solution of the winner determination problem is obvious, the matrix bids are constructed such that they are competitive. Furthermore, there is a parameter H that bounds the highest incremental value an item brings to a set. For more details on the bid types or on how the instances were generated, we refer to Day (2004).

We performed experiments on matrix bid auctions with 5, 10, 25 or 50 items and 5, 10, 25, 50, 75 or 100 bidders. For each combination, 10 instances were generated and solved to optimality. The highest incremental value per item (H) was limited to 10. We have no indication that the branch-and-price algorithm performs differently with other settings for H. All computational experiments were done on a desktop computer with a Pentium IV 2 GHz processor, with 512 MB RAM.

7.4.2 Results

Tables 7.1 and 7.2 give an overview of the average computation times needed to solve the matrix bid auction winner determination problem using branch-and-price with branching on an item-bidder pair (BOI) and branch-and-price with branching on a pair of successive of items (BOS) respectively. In Table 7.3, we give the average computation times that resulted from solving the assignment based formulation (7.1)-(7.5) with the Ilog Cplex 8.1 branch-and-cut algorithm with standard settings (B&C), which is basically the approach followed in Day & Raghavan (2006). Horizontally, the number of bidders n varies from 5 to 100, while the number of items m auctioned ranges from 5 to 50 vertically. All computation times are expressed in seconds.

As could be expected, the computation time is determined more by the number of items in the auction, than by the number of bidders. All instances with up to 10 items are solved in less than a second by all algorithms; here the branch-and-price algorithms clearly perform better. Auctions with 50 items are also solved in less than 20 minutes on average by all algorithms. The branch-and-cut algorithm seems on average the fastest way to solve these instances. Perhaps surprisingly, for the branch-and-price algorithms, the computation times for the 25 and 50 item instances do not always increase when more bidders come into play. This can be explained by the fact that the computation times for the individual instances tend to vary considerably.

One way to get a more accurate view on what the underlying trend is, is to consider a larger sample set. Also, it is not uncommon in literature on combinatorial auctions to study the median instead (see for instance Sandholm et al. (2005) and Hoos & Boutilier (2000)). Tables 7.4 to 7.6 give an overview of the median computation times needed to solve the winner determination problem. The tables shows a clear trend of how the computation times rise with the number of bidders and the number of items, since the median is

	$n = 5$	10	25	50	75	100
$m = 5$	0.005	0.007	0.008	0.017	0.027	0.038
10	0.027	0.038	0.053	0.088	0.118	0.169
25	0.636	0.597	1.157	4.292	12.704	49.155
50	247.224	60.711	437.951	557.083	622.591	802.483

Table 7.1: Average computation times [s] for n bidders and m items using BOI

	$n = 5$	10	25	50	75	100
$m = 5$	0.005	0.006	0.006	0.018	0.027	0.038
10	0.033	0.037	0.044	0.067	0.104	0.182
25	0.698	0.767	1.194	3.814	16.300	97.122
50	76.598	67.584	843.435	259.079	645.632	983.539

Table 7.2: Average computation times [s] for n bidders and m items using BOS

	$n = 5$	10	25	50	75	100
$m = 5$	0.030	0.027	0.049	0.052	0.070	0.102
10	0.050	0.069	0.140	0.278	0.524	0.748
25	0.757	1.391	3.598	10.689	17.584	31.940
50	57.676	28.333	91.230	215.083	355.785	811.960

Table 7.3: Average computation times [s] for n bidders and m items using B&C

less affected by extreme values. It is also confirmed that the branch-and-price algorithms manage to solve the majority of the instances with many items a lot faster than reflected by the average computation times. The branch-and-cut algorithm seems to suffer less from instances with extreme computation times, since the median computation time is much closer to the average computation time. The results show that computation times for the branch-and-price algorithm with branching on an item-bidder pair rise more severely with an increasing number of items than those of the branch-and-cut algorithm. On the other hand, the branch-and-price algo-

	$n = 5$	10	25	50	75	100
$m = 5$	0.000	0.010	0.010	0.020	0.030	0.040
10	0.015	0.020	0.055	0.055	0.105	0.130
25	0.480	0.460	0.760	2.445	9.480	16.825
50	20.855	29.105	45.605	129.870	227.370	353.970

Table 7.4: Median computation times [s] for n bidders and m items using BOI

	$n = 5$	10	25	50	75	100
$m = 5$	0.000	0.010	0.010	0.020	0.030	0.040
10	0.015	0.020	0.040	0.055	0.100	0.130
25	0.485	0.495	0.815	2.445	6.790	13.605
50	20.855	29.215	37.970	129.870	238.785	514.370

Table 7.5: Median computation times [s] for n bidders and m items using BOS

	$n = 5$	10	25	50	75	100
$m = 5$	0.020	0.025	0.040	0.050	0.070	0.105
10	0.040	0.060	0.140	0.260	0.535	0.740
25	0.530	1.235	3.245	10.595	18.120	28.960
50	14.665	22.615	73.035	191.670	350.340	589.940

Table 7.6: Median computation times [s] for n bidders and m items using B&C

rithm with branching on an item-bidder pair handles an increasing number of bidders better than the branch-and-cut algorithm. Furthermore, apart from a couple of exceptions, the median computation times are lower with branch-and-price than with branch-and-cut.

Tables 7.7 and 7.8 give the average computation times for solving the LP-relaxation of the set packing formulation (7.6)-(7.9) and the assignment formulation (7.1)-(7.5) respectively. Recall that the former is used in both branch-and-price algorithms, while the latter is used in the branch-and-cut

	$n = 5$	10	25	50	75	100
$m = 5$	0.01 [8]	0.01 [8]	0.01 [7]	0.02 [9]	0.03 [10]	0.04 [10]
10	0.01 [6]	0.02 [9]	0.03 [4]	0.05 [8]	0.09 [8]	0.14 [7]
25	0.31 [3]	0.41 [7]	0.80 [8]	2.12 [8]	3.29 [5]	4.71 [4]
50	14.02 [5]	14.96 [6]	32.93 [4]	74.46 [7]	124.29 [5]	116.19 [4]

Table 7.7: Average computation times [s] for the LP-relaxation of the set packing formulation for n bidders and m items

	$n = 5$	10	25	50	75	100
$m = 5$	0.01 [8]	0.02 [9]	0.02 [8]	0.03 [8]	0.04 [10]	0.04 [9]
10	0.02 [6]	0.03 [9]	0.05 [5]	0.09 [7]	0.14 [7]	0.20 [8]
25	0.28 [5]	0.51 [8]	1.61 [7]	4.47 [9]	6.03 [5]	8.35 [5]
50	9.40 [5]	20.01 [6]	43.93 [3]	159.46 [6]	313.76 [4]	461.91 [4]

Table 7.8: Average computation times [s] for the LP-relaxation of the assignment formulation for n bidders and m items

algorithm. The tables might be influenced by the fact that the LP-relaxation of both formulations can be solved in polynomial time. Furthermore, with two exceptions, the LP-relaxation of the set packing formulation is solved faster than the the LP-relaxation of the assignment formulation. Between brackets, the number of instances out of 10 for which the LP-relaxation resulted in an integral solution is indicated. Notice that Theorem 24 does not imply that these numbers should be at least as high for the assignment formulation than for the set packing formulation. Indeed, if there exists an integral optimal solution, the algorithms may not find it as there may be fractional solutions with the same objective value. Further, the number of instances for which an integral optimal solution was found remains more or less constant over the bidders, while it drops for instances with more items. Not surprisingly, instances with an integral LP-relaxation have low computation times. Therefore, the figures in Table 7.7 partially explain the fluctuations in average computation times for the instances with 25 or 50 items (see Tables 7.1 and 7.2).

Finally, Table 7.9 gives an overview of the performance details of the three algorithms. Column A gives the average number of nodes in the branching tree that were explored. Column B represents the average number of pricing rounds, and column C gives the average number of variables that were generated (these columns are not applicable for the branch-and-cut algorithm). On the rows, we find the instances, where the first number indicates the number of items and the second gives the number of bidders. There seems to be no systematic difference between the branch-and-price algorithms for any of the three parameters described in this table. The branch-and-cut algorithm solves very little nodes in its branching tree, compared to the branch-and-price algorithms. In many cases, the branch-and-cut algorithm prefers generating valid inequalities in the root node to branching.

7.5 Conclusion

In this chapter, we studied the winner determination problem for the matrix bid auction. We first looked at a special case of the matrix bid auction, namely where all bidders have an identical ranking of the items. For this auction, the winner determination problem is still NP-hard, although there exists a polynomial time algorithm in the case the number of bidders is fixed. Then, we compared two mathematical formulations for the winner determination problem of the general matrix bid auction. One assignment is based on the assignment problem, while the other is based on the set packing problem. We found that both formulations are equally strong. Moreover, an integral solution for one formulation can always be translated to an integral solution for the other formulation. We used the set packing formulation as a basis for a column generation approach where the pricing problem can be solved as a shortest path problem. This means that we are able to solve the LP relaxation of the set packing formulation in polynomial time. We

	BOI			**BOS**			**B&C**
Inst.	A	B	C	A	B	C	A
5-5	2.2	3.9	33.3	2.4	4.7	34.4	1.0
5-10	1.3	3.7	68.4	1.4	3.4	68.3	1.0
5-25	2.5	4.5	142.2	1.6	3.2	141.6	1.0
5-50	1.3	3.2	266.9	1.2	3.1	266.9	1.0
5-75	1.0	2.4	435.9	1.0	2.4	435.9	1.0
5-100	1.0	2.0	565.0	1.0	2.0	565.0	1.0
10-5	7.6	18.3	124.2	9.4	27.5	93.7	1.2
10-10	7.6	16.5	201.8	5.4	16.8	193.8	1.5
10-25	4.9	10.2	363.8	2.8	8.4	352.0	1.0
10-50	3.7	8.0	788.9	1.8	5.4	782.2	1.0
10-75	2.3	6.9	1,117.7	1.6	6.0	1,113.8	1.0
10-100	2.3	7.0	1,459.5	2.6	8.4	1,455.4	1.0
25-5	7.7	72.3	1,017.7	6.4	89.4	723.1	1.2
25-10	2.0	37.8	990.7	6.8	51.7	864.5	1.5
25-25	4.4	31.2	1,793.3	3.8	33.2	1,752.1	1.0
25-50	8.6	59.7	3,703.2	5.2	55.1	3,602.4	1.0
25-75	30.0	123.5	5,402.7	32.8	143.1	5,412.7	1.0
25-100	96.8	349.3	7,564.0	163.0	635.3	7,895.2	1.3
50-5	21.4	3,095.8	4,745.9	37.9	1,279.2	2,872.6	11.8
50-10	12.2	592.6	3,963.3	27.9	611.7	4,112.8	1.0
50-25	315.1	1,494.0	11,752.4	1,029.6	2,806.7	10,141.0	1.2
50-50	361.6	938.5	16,278.1	67.6	468.4	14,359.3	1.0
50-75	102.5	828.5	20,773.4	106.7	852.0	20,776.0	1.0
50-100	96.0	839.5	30,538.0	100.4	995.4	31,120.3	5.7

Table 7.9: Performance details for the three algorithms (BOI, BOS, B&C)

then extended this approach to two branch-and-price algorithms. In one algorithm, we branch on the items, while in the other, branching is done on the succession of the items. The pricing problem for these branch-and-price algorithms remains solvable as a shortest path problem throughout the search tree. These algorithms are tested on randomly generated instances with up to 50 items and 100 bidders, which they solved within 20 minutes (on average). Finally, the branch-and-price algorithms withstood the comparison with a branch-and-cut algorithm, based on Day & Raghavan (2006). The algorithms perform better on instances with up to 10 items, but are outperformed by the branch-and-cut algorithm on some of the larger instances. The increase in computation time, however, seems favorable for the branch-and-price algorithms, which indicates that they form at least a viable approach to solve instances of the matrix bid auction winner determination problem.

Chapter 8

Topics for future research

To conclude this thesis, we elaborate on a number of topics for future research. In section 8.1, we discuss potential improvements for solving the winner determination problem of the total quantity discount auction, and in section 8.2, we present some future research topics for the matrix bid auction. All computational experiments in this chapter were done on a desktop computer with a Pentium *IV* 2 GHz processor, with 512 MB RAM.

8.1 The total quantity discount auction

In chapters 2 to 4 we discussed the total quantity discount auction. With respect to the computational side of this research, the main result was that the winner determination problem allowed a formulation with an LP-relaxation that can be solved by solving a min-cost flow problem. Moreover, we were able to generalize this result to a setting in which a number of bids are accepted beforehand (see section 2.5.2). This allowed us to develop a branch-and-bound algorithm where every node of the branching tree is be solved as a min-cost flow problem (see section 4.2).

Given the efficiency with which a min-cost flow problem can be solved, it should not be surprising that the min-cost flow branch-and-bound algorithm solves a node of its branching tree on average 10 times faster than the linear programming based branch-and-bound algorithm, and up to 2000 times faster than the branch-and-cut algorithm. This advantage makes the min-cost flow based algorithm the fastest algorithm on all instances with up to 20 bidders. However, for the instances with 50 bidders, the min-cost flow algorithm is outperformed by the LP-based branch-and-bound algorithm, and even more by the branch-and-cut algorithm. For these instances, the min-cost flow approach needs to solve 10 to 60 times more nodes than the LP-based approach. In other words, the min-cost flow approach loses the advantage it has in terms of node solution time because of an inefficient branching strategy.

Variable selection	Computation time [s]	Number of nodes
Highest priority	59.55	65,241
Min. integer feas.	983.12	240,464
Max. integer feas.	1,984.41	419,591
Pseudo costs	25.11	5,882
Strong branching	125.75	1,869
Pseudo reduced costs	61.73	15,352

Table 8.1: Average computation times [s] and number of nodes for various variable selection strategies in a depth-first search

The node selection strategy that is used in the min-cost flow based algorithm is depth-first search. To determine the variable on which to branch, we determine the highest priority bidder as the bidder for which the number of volume intervals minus the index of the LP-interval is maximal. The LP-interval is then the first interval to fix, followed by the interval directly above this interval, the interval directly below, and so on (see section 4.2). This highest priority strategy is not readily available in Ilog Cplex, which presents minimum integer feasibility, maximum integer feasibility, pseudo costs, strong branching, and pseudo reduced costs as its variable selection strategies. Table 8.1 gives an overview of the performance of these strategies in a depth-first search, as tested on (structured and random) instances with 50 bidders, 100 items, and at most 3 bids per bidder. In terms of number of nodes that need to be searched, strong branching seems the best choice. However, in terms of computation time, a pseudo cost variable selection strategy is the better option. In conclusion, this table suggests that plugging in a pseudo costs rule in the min-cost flow algorithm could decrease the number of nodes in the branching tree substantially, possibly improving the computation times by a factor of 10.

Node selection	Computation time [s]	Number of nodes
Depth-first search	25.11	5,882
Best-bound search	14.37	3,300
Best-estimate search	14.51	3,282
Alternate best-estimate search	15.25	3,385

Table 8.2: Average computation times [s] and number of nodes for various search strategies using a pseudo costs variable selection rule

Furthermore, Ilog Cplex presents other search strategies than depth-first search, which was used in the min-cost flow based algorithm. Assuming a pseudo costs variable selection rule, the performance of these search strategies on instances with 50 bidders, 100 items, and at most 3 bids per bidder is represented in Table 8.2. The table points to best-bound search as the best strategy. Furthermore, the table suggests that implementing this search strategy in the min-cost flow approach can lead to an additional decrease in computation time of over 40%.

Implementing the min-cost flow based branch-and-bound algorithm with a pseudo costs variable selection rule and a best-bound search strategy, and verifying whether and to what extent this improves the computation times remains a topic for further research.

8.2 The matrix bid auction

In chapter 7, we developed two branch-and-price algorithms for the winner determination problem of the matrix bid auction. We tested the performance of these algorithms on instances with various numbers of bidders and items. Each instance consists of matrix bids that were randomly chosen out of seven possible types: additive preference bids, single-minded bids, nested flat bids, nested k-of bids, partition bids, add-on bids, and a bid type that has non-increasing rows and columns (see also section 6.2 and Day (2004)). The question arises though how the algorithms would perform on instances that are composed out of only one single bid type. To this end we generated 10 instances for every bid type with 10 bidders and 50 items. We also investigated a random bid type, where the entries in the matrix bid are simply randomly picked numbers between 0 and 10.

Table 8.3 gives an overview of the average computation time needed to solve instances with bids of the mentioned bid types, using branch-and-price with branching on an item-bidder pair (BOI), branch-and-price with branching on a pair of successive of items (BOS), and the Ilog Cplex 8.1 branch-and-cut algorithm with standard settings (B&C). The last line in the table repeats the computation times mentioned in section 7.4 for instances with 50 items and 10 bidders and a combination of various bid types. The table clearly shows that the difficulty of the various bid types is diverse. Additive preference bids, single minded bids, nested flat bids and non-increasing bids are indeed solved much faster than the other bid types. For most of these bid types, branch-and-price with branching on a pair of successive of items is the fastest algorithm. On the other hand, the instances with nested k-of bids and

Bid type	BOI	BOS	B&C
Additive preference bid	12.71	7.88	5.91
Single-minded bid	0.13	0.13	9.75
Nested flat bid	0.14	0.14	12.11
Nested k-of bid	*	*	*
Partition bid	*	*	61.11
Add-on bid	73.49	85.34	82.14
Non-increasing bid	0.43	0.43	2.94
Random bid	81.72	106.43	282.02
Mixed bid types	60.71	67.58	28.33

Table 8.3: Average computation times [s] for various bid types (*not all instances could be solved)

partition bids are a hard nut to crack for the branch-and-price algorithms, since not all instances could be solved (due to a lack of memory). The latter bid type turns out to be very difficult for the branch-and-cut algorithm as well, since for some instances, even over 80 hours of computation time did not suffice to solve them to optimality. Finally, instances with add-on bids and random bids also require quite some computation time. For these instances, branch-and-price with branching on an item-bidder pair performs best; especially for the random instances, the difference with branch-and-cut is considerable.

Finally, Table 8.4 gives an overview of the performance details of the branch-and-price algorithms. Column A gives the average number of nodes in the branching tree that were explored. Column B represents the average number of pricing rounds, and column C gives the average number of variables that were generated. The table supports the differences in computation time for the various bid types that are mentioned in Table 8.3.

	BOI			**BOS**		
Bid type	A	B	C	A	B	C
Additive preference bid	3.3	52.9	11,752	2.3	24.8	12,111
Single-minded bid	1.0	1.3	11	1.0	1.3	11
Nested flat bid	1.0	1.3	292	1.0	1.3	292
Add-on bid	17.3	813.4	2,312	46.6	789.9	5,134
Non-increasing bid	1.0	4.4	203	1.0	4.4	203
Random bid	24.0	383.9	13,127	117.0	447.4	14,881
Mixed bid types	12.2	592.6	3,963	27.9	611.7	4,113

Table 8.4: Performance details for the branch-and-price algorithms (BOI, BOS)

Another issue is that the bid functions that can be represented in a single matrix bid seem restrictive. Therefore, it would be interesting to extend the matrix bid auction to a setting where multiple matrix bids per bidder can be submitted. However, we would have to ensure that at most one bid per bidder can be accepted. For the branch-and-price algorithm, this could be done as follows. We create a dummy item for each bidder. This dummy item is to be inserted in each matrix bid of this bidder as the highest ranked item. The first column of each matrix bid has a zero in the entry corresponding to the dummy item, and a highly negative value for all other entries. Obviously, each positive bid will have to include the dummy item, and thus the constraint that each item can be auctioned at most once will avoid that bids from different matrix bids by the same bidder are accepted. In this way, the branch-and-price algorithms can still be applied for this variant with multiple matrix bids per bidder.

Notice that any bid function can be represented by a series of such matrix bids, although this may require a number of matrix bids that is exponential in the number of items. Indeed, a matrix bid like the one described above could be created for each bid on a subset of items. An interesting topic for further research would be to develop a way to represent an arbitrary set of bids using as few matrix bids as possible. If this number turns out to be small, the resulting winner determination problem may still be efficiently solvable in practice.

List of Figures

List of Tables

Bibliography

Achterberg, T., Koch, T. & Martin, A. (2005). Branching rules revisited, *Operations Research Letters* **33**: 42–54.

Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A. & Protasi, M. (1999). *Complexity and approximation: combinatorial optimization problems and their approximability properties*, Springer.

Austin, L. M. & Hogan, W. W. (1976). Optimizing the procurement of aviation fuels, *Management Science* **22**(5): 515–527.

Ausubel, L. M. & Milgrom, P. (2002). Ascending auctions with package bidding, *Frontiers of Theoretical Economics* **1**: 1–42.

Ausubel, L. M. & Milgrom, P. (2005). The lovely but lonely Vickrey auction, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 17–40.

Ball, M. O., Donohue, G. L. & Hoffman, K. (2005). Auctions for the safe, efficient and equitable allocation of airspace system resources, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 507–538.

Ballou, R. H. (1998). *Business logistics management*, Prentice Hall.

Banks, J. S., Ledyard, J. O. & Porter, D. P. (1989). Allocating uncertain and unresponsive resources: An experimental approach, *RAND Journal of Economics* **20**(1): 1–25.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. & Vance, P. H. (1998). Branch-and-price: column generation for solving huge integer programs, *Operations Research* **46**: 316–329.

Beliën, J. (2006). *Exact and heuristic methodologies for scheduling in hospitals: problems, formulations and algorithms*, PhD thesis, K.U.Leuven.

Bevia, C., Quinzii, M. & Silva, J. A. (1999). Buying several indivisible goods, *Mathematical Social Sciences* **37**(1): 1–23.

Bichler, M., Kalagnanam, J., Lee, H. S. & Lee, J. (2002). Winner determination algorithms for electronic auctions: A framework design, *in EC-Web '02: Proceedings of the 3rd international conference on e-commerce and web technologies, Aix-en-Provence, France*, Springer, pp. 37–46.

Bikhchandani, S., de Vries, S., Schummer, J. & Vohra, R. (2002). Linear programming and Vickrey auctions, *in* B. Dietrich & R. Vohra (eds), *Mathematics of the Internet: e-auction and markets*, Springer, pp. 75–116.

Bikhchandani, S. & Mamer, J. (1997). Competitive equilibrium in an exchange economy with indivisibilities, *Journal of Economic Theory* **74**: 385–413.

Bleischwitz, Y. & Kliewer, G. (2005). Accelerating Vickrey payment computation in combinatorial auctions for an airline alliance., *in WEA '05: Proceedings of the 4th international workshop on experimental and efficient algorithms, Santorini Island, Greece*, Springer, pp. 228–239.

Cantillon, E. & Pesendorfer, M. (2005). Auctioning bus routes: the London experience, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 573–591.

Cao, B. (1992). Transportation problem with nonlinear side constraints a branch and bound approach, *Zeitschrift für Operations Research* **36**: 185–197.

Cao, B. & Uebe, G. (1995). Solving transportation problems with nonlinear side constraints with tabu search, *Computers and Operations Research* **22**: 593–603.

Caplice, C. & Sheffi, Y. (2005). Combinatorial auctions for truckload transportation, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 539–571.

Chauhan, S. S., Eremeev, A. V., Romanova, A. A., Servakh, V. V. & Woeginger, G. J. (2005). Approximation of the supply scheduling problem, *Operations Research Letters* **33**(3): 249–254.

Crama, Y., Pascual, J. R. & Torres, A. (2004). Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes, *European Journal of Operational Research* **159**(2): 364–378.

Cramton, P. (2002). Spectrum auctions, *in* M. Cave, S. Majumdar & I. Vogelsang (eds), *Handbook of telecommunications economics*, Elsevier, chapter 14.

Cramton, P., Steinberg, R. & Shoham, Y. (2005). *Combinatorial auctions*, MIT Press.

Dantzig, G. B. & Wolfe, P. (1960). Decomposition principle for linear programs, *Operations Research* **8**: 101–111.

Davenport, A. J. & Kalagnanam, J. R. (2002). Price negotiations for procurement of direct inputs, *in* B. Dietrich & R. Vohra (eds), *Mathematics of the Internet: e-auction and markets*, Springer, pp. 27–43.

Day, R. W. (2004). *Expressing preferences with price-vector agents in combinatorial auctions*, PhD thesis, University of Maryland.

Day, R. W. & Raghavan, S. (2006). Matrix bidding in combinatorial auctions. Manuscript.

de Vries, S., Schummer, J. & Vohra, R. V. (in press). On ascending Vickrey auctions for heterogeneous objects, *Journal of Economic Theory* .

de Vries, S. & Vohra, R. V. (2003). Combinatorial auctions: a survey, *INFORMS Journal on Computing* (3): 284–309.

Degraeve, Z., Labro, E. & Roodhooft, F. (2000). An evaluation of vendor selection models from a total cost of ownership perspective, *European Journal of Operational Research* **125**: 34–58.

Dobzinski, S., Nisan, N. & Schapira, M. (2005). Approximation algorithms for combinatorial auctions with complement-free bidders, *in STOC '05: Proceedings of the 37th annual ACM symposium on theory of computing, Baltimore, USA*, ACM Press, pp. 610–618.

Epstein, R., Henríquez, L., Catalán, J., Weintraub, G. Y. & Martínez, C. (2002). A combinational auction improves school meals in Chile, *Interfaces* **32**(6): 1–14.

Eso, M. (2001). An iterative online auction for airline seats, *in* B. Dietrich & R. Vohra (eds), *Mathematics of the Internet: e-auction and markets*, Springer-Verlag, pp. 45–58.

Eso, M., Ghosh, S., Kalagnanam, J. R. & Ladányi, L. (2005). Bid evaluation in procurement auctions with piecewise linear supply curves, *Journal of Heuristics* **11**(2): 147–173.

Fujishige, S. & Yang, Z. (2003). A note on Kelso and Crawford's gross substitutes condition, *Mathematics of Operations Research* **28**(3): 463–469.

Garey, M. R. & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman.

Garey, M. R., Tarjan, R. E. & Wilfong, G. (1988). One-processor scheduling with symmetric earliness and tardiness penalties, *Mathematics of Operations Research* **13**: 330–349.

Gomory, R. E. (1963). An algorithm for integer solutions to linear programs, *in* R. L. Graves & P.Wolfe (eds), *Recent advances in mathematical programming*, McGraw-Hill, pp. 269–302.

Goossens, D., Maas, A. J. T., Spieksma, F. C. R. & van de Klundert, J. J. (2007). Exact algorithms for procurement problems under a total quantity discount structure, *European Journal of Operational Research* **178**(2): 603–626.

Goossens, D. & Spieksma, F. C. R. (2005). The transportation problem with exclusionary side constraints, *DTEW Research Report 0565*, K.U.Leuven.

Goossens, D. & Spieksma, F. C. R. (2006a). Exact algorithms for the matrix bid auction, *FETEW Research Report KBI-0620*, K.U.Leuven.

Goossens, D. & Spieksma, F. C. R. (2006b). The matrix bid auction: micro-economic properties and expressiveness, *FETEW Research Report KBI-0619*, K.U.Leuven.

Grötschel, M., Lovász, L. & Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1**(2): 169–197.

Gul, F. & Stacchetti, E. (1999). Walrasian equilibrium with gross substitutes, *Journal of Economic Theory* **87**(1): 95–124.

Günlük, O., Ladányi, L. & de Vries, S. (2005). A branch-and-price algorithm and new test problems for spectrum auctions, *Management Science* **53**(3): 391–406.

Hohner, G., Rich, J., Ng, E., Reid, G., Davenport, A. J., Kalagnanam, J., Lee, H. S. & Chae, A. (2003). Combinatorial and quantity-discount procurement auctions benefit Mars, incorporated and its suppliers, *Interfaces* **33**(1): 23–35.

Hoos, H. & Boutilier, C. (2000). Solving combinatorial auctions using stochastic local search, *in AAAI/IAAI '00: Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence, Austin, USA*, AAAI Press/The MIT Press, pp. 22–29.

Johnson, E., Nemhauser, G. & Savelsbergh, M. (2000). Progress in linear programming based branch-and-bound algorithms: an exposition, *INFORMS Journal on Computing* **12**: 2–23.

Kann, V. (1991). Maximum bounded 3-dimensional matching is max snp-complete, *Information Processing Letters* **37**(1): 27–35.

Katz, P., Sadrian, A. A. & Tendick, P. (1994). Telephone companies analyze price quotations with Bellcore's PDSS software, *Interfaces* **24**(1): 50–63.

Kelly, F. & Steinberg, R. (2000). A combinatorial auction with multiple winners for universal service, *Management Science* **46**(4): 586–596.

Kelso, Alexander S, J. & Crawford, V. P. (1982). Job matching, coalition formation, and gross substitutes, *Econometrica* **50**(6): 1483–1504.

Khot, S., Lipton, R. J., Markakis, E. & Mehta, A. (2005). Inapproximability results for combinatorial auctions with submodular utility functions, *in WINE '05: Proceedings of the 1st international work-*

shop on Internet and network economics, Hong Kong, China, Springer, pp. 92–101.

Kim, D. & Pardalos, P. M. (2001). Piecewise linear network flow problems, *in* P. M. Pardalos, A. Migdalas & R. Burkhard (eds), *Combinatorial and global optimization*, World Scientific, pp. 145–160.

Klemperer, P. (2000). *The economic theory of auctions*, Edward Elgar.

Kothari, A., Parkes, D. C. & Suri, S. (2003). Approximately-strategyproof and tractable multi-unit auctions, *in EC '03: Proceedings of the 4th ACM conference on electronic commerce, San Diego, USA*, ACM Press, pp. 166–175.

Lasdon, L. (1970). *Optimization theory for large systems*, MacMillan.

Ledyard, J. O., Olson, M., Porter, D., Swanson, J. A. & Torma, D. P. (2002). The first use of a combined-value auction for transportation services, *Interfaces* **32**(5): 4–12.

Lehmann, B., Lehmann, D. & Nisan, N. (2001). Combinatorial auctions with decreasing marginal utilities, *in EC '01: Proceedings of the 3rd ACM conference on electronic commerce, Tampa, USA*, ACM Press, pp. 18–28.

Leyton-Brown, K., Nudelman, E. & Shoham, Y. (2005). Empirical hardness models for combinatorial auctions, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 479–504.

Leyton-Brown, K. & Shoham, Y. (2005). A test suite for combinatorial auctions, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 451–478.

Leyton-Brown, K., Shoham, Y. & Tennenholtz, M. (2000). An algorithm for multi-unit combinatorial auctions, *in AAAI/IAAI '00:*

Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence, Austin, USA, AAAI Press / The MIT Press, pp. 56–61.

Linderoth, J. & Savelsbergh, M. (1999). A computational study of search strategies for mixed integer programming, *INFORMS Journal on Computing* **11**: 173–187.

Martello, S. & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*, John Wiley and Sons Ltd.

McConnel, M. B. & Galligan, D. T. (2004). The use of integer programming to select bulls across breeding companies with volume discounts, *Journal of Diary Science* **87**: 3542–3549.

Milgrom, P. (2004). *Putting auction theory to work*, Cambridge University Press.

Mitchell, J. E. (2002). Branch-and-cut algorithms for combinatorial optimization problems, *in* P. M. Pardalos & M. G. C. Resende (eds), *Handbook of applied optimization*, Oxford University Press, pp. 65–77.

Moonen, L. (2005). *Algorithms for some graph-theoretical optimization problems*, PhD thesis, K.U.Leuven.

Moulin, H. (1988). *Axioms of cooperative decision making*, Cambridge University Press.

Müller, R. (2005). Tractable cases of the winner determination problem, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 319–336.

Munson, C. L. & Rosenblatt, M. J. (1998). Theories and realities of quantity discounts: an explanatory study, *Production and Operations Management* **7**(4): 352–369.

Murota, K. & Tamura, A. (2003). New characterizations of m-convex functions and their applications to economic equilibrium models with indivisibilities, *Discrete Applied Mathematics* **131**(2): 495–512.

Nemhauser, G. L. & Wolsey, L. A. (1988). *Integer and combinatorial optimization*, Wiley.

Nisan, N. (2000). Bidding and allocation in combinatorial auctions, *in EC '00: Proceedings of the 2nd ACM conference on electronic commerce, Minneapolis, USA*, ACM Press, pp. 1–12.

Nisan, N. (2005). Bidding languages for combinatorial auctions, *in* P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 215–231.

Nisan, N. & Segal, I. (2006). The communication requirements of efficient allocations and supporting prices, *Journal of Economic Theory* **129**(1): 192–224.

Park, S. & Rothkopf, M. H. (2005). Auctions with bidder-determined allowable combinations, *European Journal of Operational Research* **161**(2): 399–415.

Parkes, D. C. (1999). iBundle: an efficient ascending price bundle auction, *in EC '99: Proceedings of the ACM conference on electronic commerce, Denver, USA*, ACM Press, pp. 148–157.

Parkes, D. C. & Ungar, L. H. (2000). Iterative combinatorial auctions: theory and practice, *in AAAI/IAAI '00: Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence, Austin, USA*, AAAI Press/The MIT Press, pp. 74–81.

Petrank, E. (1994). The hardness of approximation: gap location, *Computational Complexity* **4**(2): 133–157.

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

Reijnierse, H., Potters, J. A. & van Gellekom, A. (2002). Verifying gross substitutability, *Economic Theory* **20**(4): 767–776.

Rothkopf, M., Pekeç, A. & Harstad, R. M. (1998). Computationally manageable combinational auctions, *Management Science* **44**(8): 1131–1147.

Ryan, D. & Foster, B. (1981). An integer programming approach to scheduling, *in* A. Wren (ed.), *Computer scheduling of public transport: urban passenger vehicle and crew scheduling*, North-Holland, pp. 269–280.

Sadrian, A. A. & Yoon, Y. S. (1994). A procurement decision support system in business volume discount environments, *Operations Research* **42**(1): 14–23.

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

Sandholm, T., Suri, S., Gilpin, A. & Levine, D. (2002). Winner determination in combinatorial auction generalizations, *in AAMAS '02: Proceedings of the 1st international joint conference on autonomous agents and multiagent systems, Bologna, Italy*, ACM Press, pp. 69–76.

Sandholm, T., Suri, S., Gilpin, A. & Levine, D. (2005). CABOB: a fast optimal algorithm for combinatorial auctions, *Management Science* **51**: 374–390.

Shachnai, H., Shmueli, O. & Sayegh, R. (2004). Approximation schemes for deal splitting and covering integer programs with multiplicity constraints, *in WAOA '04: Proceedings of the 2nd international workshop* *on approximation and online algorithms, Bergen, Norway*, Springer, pp. 111–125.

Sheffi, Y. (2004). Combinatorial auctions in the procurement of transportation services, *Interfaces* **34**(4): 245–252.

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

Sun, M. (2002). The transportation problem with exclusionary side constraints and two branch-and-bound algorithms, *European Journal of Operational Research* **140**(3): 629–647.

Syarif, A. & Gen, M. (2003). Solving exclusionary side constrained transportation problem by using a hybrid spanning tree-based genetic algorithm, *Journal of Intelligent Manufacturing* **14**: 389–399.

Tennenholtz, M. (2002). Tractable combinatorial auctions and b-matching, *Artificial Intelligence* **140**(1/2): 231–243.

van de Klundert, J. J., Kuipers, J., Spieksma, F. C. R. & Winkels, M. (2005). Selecting telecommunication carriers to obtain volume discounts, *Interfaces* **35**(2): 124–132.

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

Vanderbeck, F. & Wolsey, L. (1996). An exact algorithm for ip column generation, *Operations Research Letters* **19**: 151–159.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders, *Journal of Finance* **16**: 8–37.

Winston, W. L. (2004). *Operations research: applications and algorithms*, Duxbury.

Wolsey, L. (1998). *Integer programming*, Wiley.

Wright, J. W. (1975). The change-making problem, *Journal of the Association for Computing Machinery* **22**: 125–128.

Xu, J., Lu, L. L. & Glover, F. (2000). The deterministic multi-item dynamic lot size problem with joint business volume discount, *Annals of Operations Research* **96**(1): 317–337.

Doctoral Dissertations from the Faculty of Economic and Applied Economic Sciences

From August 1, 1971.

1. GEPTS Stefaan (1971)
   Stability and efficiency of resource allocation processes in discrete commodity spaces. Leuven, KUL, 1971. 86 pp.

2. PEETERS Theo (1971)
   Determinanten van de internationale handel in fabrikaten. Leuven, Acco, 1971. 290 pp.

3. VAN LOOY Wim (1971)
   Personeelsopleiding: een onderzoek naar investeringsaspekten van opleiding. Hasselt, Vereniging voor wetenschappelijk onderzoek in Limburg, 1971. VII, 238 pp.

4. THARAKAN Mathew (1972)
   Indian exports to the European community: problems and prospects. Leuven, Faculty of economics and applied economics, 1972. X,343 pp.

5. HERROELEN Willy (1972)
   Heuristische programmatie: methodologische benadering en praktische toepassing op complexe combinatorische problemen. Leuven, Aurelia scientifica, 1972. X, 367 pp.

6. VANDENBULCKE Jacques (1973)
   De studie en de evaluatie van data-organisatiemethodes en data-zoekmethodes. Leuven, s.n., 1973. 3 V.

7. PENNYCUICK Roy A. (1973)
   The economics of the ecological syndrome. Leuven, Acco, 1973. XII, 177 pp.

8. KAWATA T. Bualum (1973)
   Formation du capital d'origine belge, dette publique et stratégie du développement au Zaire. Leuven, KUL, 1973. V, 342 pp.

9. DONCKELS Rik (1974)
   Doelmatige oriëntering van de sectorale subsidiepolitiek in België: een theoretisch onderzoek met empirische toetsing. Leuven, K.U.Leuven, 1974. VII, 156 pp.

10. VERHELST Maurice (1974)
Contribution to the analysis of organizational information systems and their financial benefits. Leuven, K.U.Leuven, 1974. 2 V.

11. CLEMEUR Hugo (1974)
Enkele verzekeringstechnische vraagstukken in het licht van de nutstheorie. Leuven, Aurelia scientifica, 1974. 193 pp.

12. HEYVAERT Edward (1975)
De ontwikkeling van de moderne bank- en krediettechniek tijdens de zestiende en zeventiende eeuw in Europa en te Amsterdam in het bijzonder. Leuven, K.U.Leuven, 1975. 186 pp.

13. VERTONGHEN Robert (1975)
Investeringscriteria voor publieke investeringen: het uitwerken van een operationele theorie met een toepassing op de verkeersinfrastructuur. Leuven, Acco, 1975. 254 pp.

14. Niet toegekend.

15. VANOVERBEKE Lieven (1975)
Microeconomisch onderzoek van de sectoriële arbeidsmobiliteit. Leuven, Acco, 1975. 205 pp.

16. DAEMS Herman (1975)
The holding company: essays on financial intermediation, concentration and capital market imperfections in the Belgian economy. Leuven, K.U.Leuven, 1975. XII, 268 pp.

17. VAN ROMPUY Eric (1975)
Groot-Brittannië en de Europese monetaire integratie: een onderzoek naar de gevolgen van de Britse toetreding op de geplande Europese monetaire unie. Leuven, Acco, 1975. XIII, 222 pp.

18. MOESEN Wim (1975)
Het beheer van de staatsschuld en de termijnstructuur van de intrestvoeten met een toepassing voor België. Leuven, Vander, 1975. XVI, 250 pp.

19. LAMBRECHT Marc (1976)
Capacity constrained multi-facility dynamic lot-size problem. Leuven, KUL, 1976. 165 pp.

20. RAYMAECKERS Erik (1976)
De mens in de onderneming en de theorie van het producenten-gedrag: een bijdrage tot transdisciplinaire analyse. Leuven, Acco, 1976. XIII, 538 pp.

21. TEJANO Albert (1976)
Econometric and input-output models in development planning: the case of the Philippines. Leuven, KUL, 1976. XX, 297 pp.

22. MARTENS Bernard (1977)
Prijsbeleid en inflatie met een toepassing op België. Leuven, KUL, 1977. IV, 253 pp.

23. VERHEIRSTRAETEN Albert (1977)
Geld, krediet en intrest in de Belgische financiële sector. Leuven, Acco, 1977. XXII, 377 pp.

24. GHEYSSENS Lieven (1977)
International diversification through the government bond market: a risk-return analysis. Leuven, s.n., 1977. 188 pp.

25. LEFEBVRE Chris (1977)
Boekhoudkundige verwerking en financiële verslaggeving van huurkooptransacties en verkopen op afbetaling bij ondernemingen die aan consumenten verkopen. Leuven, KUL, 1977. 228 pp.

26. KESENNE Stefan (1978)
Tijdsallocatie en vrijetijdsbesteding: een econometrisch onderzoek. Leuven, s.n., 1978. 163 pp.

27. VAN HERCK Gustaaf (1978)
Aspecten van optimaal bedrijfsbeleid volgens het marktwaardecriterium: een risico-rendementsanalyse. Leuven, KUL, 1978. IV, 163 pp.

28. VAN POECK Andre (1979)
World price trends and price and wage development in Belgium: an investigation into the relevance of the Scandinavian model of inflation for Belgium. Leuven, s.n., 1979. XIV, 260 pp.

29. VOS Herman (1978)
De industriële technologieverwerving in Brazilië: een analyse. Leuven, s.n., 1978. onregelmatig gepagineerd.

30. DOMBRECHT Michel (1979)
Financial markets, employment and prices in open economies. Leuven, KUL, 1979. 182 pp.

31. DE PRIL Nelson (1979)
Bijdrage tot de actuariële studie van het bonus-malussysteem. Brussel, OAB, 1979. 112 pp.

32. CARRIN Guy (1979)
Economic aspects of social security: a public economics approach. Leuven, KUL, 1979. onregelmatig gepagineerd

33. REGIDOR Baldomero (1979)
An empirical investigation of the distribution of stock-market prices and weak-form efficiency of the Brussels stock exchange. Leuven, KUL, 1979. 214 pp.

34. DE GROOT Roger (1979)
Ongelijkheden voor stop loss premies gebaseerd op E.T. systemen in het kader van de veralgemeende convexe analyse. Leuven, KUL, 1979. 155 pp.

35. CEYSSENS Martin (1979)
On the peak load problem in the presence of rationizing by waiting. Leuven, KUL, 1979. IX, 217 pp.

36. ABDUL RAZK Abdul (1979)
Mixed enterprise in Malaysia: the case study of joint venture between Malysian public corporations and foreign enterprises. Leuven, KUL, 1979. 324 pp.

37. DE BRUYNE Guido (1980)
Coordination of economic policy: a game-theoretic approach. Leuven, KUL, 1980. 106 pp.

38. KELLES Gerard (1980)
Demand, supply, price change and trading volume on financial markets of the matching-order type. = Vraag, aanbod, koersontwikkeling en omzet op financiële markten van het Europese type. Leuven, KUL, 1980. 222 pp.

39. VAN EECKHOUDT Marc (1980)
De invloed van de looptijd, de coupon en de verwachte inflatie op het opbrengstverloop van vastrentende finaciële activa. Leuven, KUL, 1980. 294 pp.

40. SERCU Piet (1981)
Mean-variance asset pricing with deviations from purchasing power parity. Leuven, s.n., 1981. XIV, 273 pp.

41. DEQUAE Marie-Gemma (1981)
Inflatie, belastingsysteem en waarde van de onderneming. Leuven, KUL, 1981. 436 pp.

42. BRENNAN John (1982)
An empirical investigation of Belgian price regulation by prior notification: 1975 - 1979 - 1982. Leuven, KUL, 1982. XIII, 386 pp.

43. COLLA Annie (1982)
Een econometrische analyse van ziekenhuiszorgen. Leuven, KUL, 1982. 319 pp.

44. Niet toegekend.

45. SCHOKKAERT Eric (1982)
Modelling consumer preference formation. Leuven, KUL, 1982. VIII, 287 pp.

46. DEGADT Jan (1982)
Specificatie van een econometrisch model voor vervuilingsproblemen met proeven van toepassing op de waterverontreiniging in België. Leuven, s.n., 1982. 2 V.

47. LANJONG Mohammad Nasir (1983)
A study of market efficiency and risk-return relationships in the Malaysian capital market. s.l., s.n., 1983. XVI, 287 pp.

48. PROOST Stef (1983)
De allocatie van lokale publieke goederen in een economie met een centrale overheid en lokale overheden. Leuven, s.n., 1983. onregelmatig gepagineerd.

49. VAN HULLE Cynthia (1983)
Shareholders' unanimity and optimal corporate decision making in imperfect capital markets. s.l., s.n., 1983. 147 pp. + appendix.

50. VAN WOUWE Martine (2/12/83)
Ordening van risico's met toepassing op de berekening van ultieme ruïnekansen. Leuven, s.n., 1983. 109 pp.

51. D'ALCANTARA Gonzague (15/12/83)
SERENA: a macroeconomic sectoral regional and national account econometric model for the Belgian economy. Leuven, KUL, 1983. 595 pp.

52. D'HAVE Piet (24/02/84)
De vraag naar geld in België. Leuven, KUL, 1984. XI, 318 pp.

53. MAES Ivo (16/03/84)
The contribution of J.R. Hicks to macro-economic and monetary theory. Leuven, KUL, 1984. V, 224 pp.

54. SUBIANTO Bambang (13/09/84)
A study of the effects of specific taxes and subsidies on a firms' R&D investment plan. s.l., s.n., 1984. V, 284 pp.

55. SLEUWAEGEN Leo (26/10/84)
Location and investment decisions by multinational enterprises in Belgium and Europe. Leuven, KUL, 1984. XII, 247 pp.

56. GEYSKENS Erik (27/03/85)
Produktietheorie en dualiteit. Leuven, s.n., 1985. VII, 392 pp.

57. COLE Frank (26/06/85)
Some algorithms for geometric programming. Leuven, KUL, 1985. 166 pp.

58. STANDAERT Stan (26/09/86)
A study in the economics of repressed consumption. Leuven, KUL, 1986. X, 380 pp.

59. DELBEKE Jos (03/11/86)
Trendperioden in de geldhoeveelheid van België 1877-1983: een theoretische en empirische analyse van de "Banking school"hypothese. Leuven, KUL, 1986. XII, 430 pp.

60. VANTHIENEN Jan (08/12/86)
Automatiseringsaspecten van de specificatie, constructie en manipulatie van beslissingstabellen. Leuven, s.n., 1986. XIV, 378 pp.

61. LUYTEN Robert (30/04/87)
A systems-based approach for multi-echelon production/inventory systems. s.l., s.n., 1987. 3V.

62. MERCKEN Roger (27/04/87)
De invloed van de data base benadering op de interne controle. Leuven, s.n., 1987. XIII, 346 pp.

63. VAN CAYSEELE Patrick (20/05/87)
Regulation and international innovative activities in the pharmaceutical industry. s.l., s.n., 1987. XI, 169 pp.

64. FRANCOIS Pierre (21/09/87)
De empirische relevantie van de independence from irrelevant alternatives. Assumptie indiscrete keuzemodellen. Leuven, s.n., 1987. IX, 379 pp.

65. DECOSTER André (23/09/88)
Family size, welfare and public policy. Leuven, KUL. Faculteit Economische en Toegepaste Economische Wetenschappen, 1988. XIII, 444 pp.

66. HEIJNEN Bart (09/09/88)
Risicowijziging onder invloed van vrijstellingen en herverzekeringen: een theoretische analyse van optimaliteit en premiebepaling. Leuven, KUL. Faculteit Economische en Toegepaste Economische Wetenschappen, 1988. onregelmatig gepagineerd.

67. GEEROMS Hans (14/10/88)
Belastingvermijding. Theoretische analyse van de determinanten van de belastingontduiking en de belastingontwijking met empirische verificaties. Leuven, s.n., 1988. XIII, 409, 5 pp.

68. PUT Ferdi (19/12/88)
Introducing dynamic and temporal aspects in a conceptual (database) schema. Leuven, KUL. Faculteit Economische en Toegepaste Economische Wetenschappen, 1988. XVIII, 415 pp.

69. VAN ROMPUY Guido (13/01/89)
A supply-side approach to tax reform programs. Theory and empirical evidence for Belgium. Leuven, KUL. Faculteit Economische en Toegepaste Economische Wetenschappen, 1989. XVI, 189, 6 pp.

70. PEETERS Ludo (19/06/89)
Een ruimtelijk evenwichtsmodel van de graanmarkten in de E.G.: empirische specificatie en beleidstoepassingen. Leuven, K.U.Leuven. Faculteit Economische en Toegepaste Economische Wetenschappen, 1989. XVI, 412 pp.

71. PACOLET Jozef (10/11/89)
Marktstructuur en operationele efficiëntie in de Belgische financiële sector. Leuven, K.U.Leuven. Faculteit Economische en Toegepaste Economische Wetenschappen, 1989. XXII, 547 pp.

72. VANDEBROEK Martina (13/12/89)
Optimalisatie van verzekeringscontracten en premieberekeningsprincipes. Leuven, K.U.Leuven. Faculteit Economische en Toegepaste Economische Wetenschappen, 1989. 95 pp.

73. WILLEKENS Francois (1990)
Determinance of government growth in industrialized countries with applications to Belgium. Leuven, K.U.Leuven. Faculteit Economische en Toegepaste Economische Wetenschappen, 1990. VI, 332 pp.

74. VEUGELERS Reinhilde (02/04/90)
Scope decisions of multinational enterprises. Leuven, K.U.Leuven. Faculteit Economische en Toegepaste Economische Wetenschappen, 1990. V, 221 pp.

75. KESTELOOT Katrien (18/06/90)
Essays on performance diagnosis and tacit cooperation in international oligopolies. Leuven, K.U.Leuven. Faculteit Economische en Toegepaste Economische Wetenschappen, 1990. 227 pp.

76. WU Changqi (23/10/90) Strategic aspects of oligopolistic vertical integration. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1990. VIII, 222 pp.

77. ZHANG Zhaoyong (08/07/91)
A disequilibrium model of China's foreign trade behaviour. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1991. XII, 256 pp.

78. DHAENE Jan (25/11/91)
Verdelingsfuncties, benaderingen en foutengrenzen van stochastische grootheden geassocieerd aan verzekeringspolissen en -portefeuilles. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1991. 146 pp.

79. BAUWELINCKX Thierry (07/01/92)
Hierarchical credibility techniques. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1992. 130 pp.

80. DEMEULEMEESTER Erik (23/3/92)
Optimal algorithms for various classes of multiple resource-constrained project scheduling problems. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1992. 180 pp.

81. STEENACKERS Anna (1/10/92)
Risk analysis with the classical actuarial risk model: theoretical extensions and applications to Reinsurance. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1992. 139 pp.

82. COCKX Bart (24/09/92)
The minimum income guarantee. Some views from a dynamic perspective. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1992. XVII, 401 pp.

83. MEYERMANS Eric (06/11/92)
Econometric allocation systems for the foreign exchange market: Specification, estimation and testing of transmission mechanisms under currency substitution. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1992. XVIII, 343 pp.

84. CHEN Guoqing (04/12/92)
Design of fuzzy relational databases based on fuzzy functional dependency. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1992. 176 pp.

85. CLAEYS Christel (18/02/93)
Vertical and horizontal category structures in consumer decision making: The nature of product hierarchies and the effect of brand typicality. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 348 pp.

86. CHEN Shaoxiang (25/03/93)
The optimal monitoring policies for some stochastic and dynamic production processes. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 170 pp.

87. OVERWEG Dirk (23/04/93)
Approximate parametric analysis and study of cost capacity management of computer configurations. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 270 pp.

88. DEWACHTER Hans (22/06/93)
Nonlinearities in speculative prices: The existence and persistence of nonlinearity in foreign exchange rates. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 151 pp.

89. LIN Liangqi (05/07/93)
Economic determinants of voluntary accounting choices for R & D expenditures in Belgium. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 192 pp.

90. DHAENE Geert (09/07/93)
Encompassing: formulation, properties and testing. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 117 pp.

91. LAGAE Wim (20/09/93)
Marktconforme verlichting van soevereine buitenlandse schuld door private crediteuren: een neo-institutionele analyse. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 241 pp.

92. VAN DE GAER Dirk (27/09/93)
Equality of opportunity and investment in human capital. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1993. 172 pp.

93. SCHROYEN Alfred (28/02/94)
Essays on redistributive taxation when monitoring is costly. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1994. 203 pp. + V.

94. STEURS Geert (15/07/94)
Spillovers and cooperation in research and development. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1994. 266 pp.

95. BARAS Johan (15/09/94)
The small sample distribution of the Wald, Lagrange multiplier and likelihood ratio tests for homogeneity and symmetry in demand analysis: a Monte Carlo study. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1994. 169 pp.

96. GAEREMYNCK Ann (08/09/94)
The use of depreciation in accounting as a signalling device. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1994. 232 pp.

97. BETTENDORF Leon (22/09/94)
A dynamic applied general equilibrium model for a small open economy. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1994. 149 pp.

98. TEUNEN Marleen (10/11/94)
Evaluation of interest randomness in actuarial quantities. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1994. 214 pp.

99. VAN OOTEGEM Luc (17/01/95)
An economic theory of private donations. Leuven. K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 236 pp.

100. DE SCHEPPER Ann (20/03/95)
Stochastic interest rates and the probabilistic behaviour of actuarial functions. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 211 pp.

101. LAUWERS Luc (13/06/95)
Social choice with infinite populations. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 79 pp.

102. WU Guang (27/06/95)
A systematic approach to object-oriented business modeling. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 248 pp.

103. WU Xueping (21/08/95)
Term structures in the Belgian market: model estimation and pricing error analysis. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 133 pp.

104. PEPERMANS Guido (30/08/95)
Four essays on retirement from the labor force. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 128 pp.

105. ALGOED Koen (11/09/95)
Essays on insurance: a view from a dynamic perspective. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 136 pp.

106. DEGRYSE Hans (10/10/95)
Essays on financial intermediation, product differentiation, and market structure. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 218 pp.

107. MEIR Jos (05/12/95)
Het strategisch groepsconcept toegepast op de Belgische financiële sector. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1995. 257 pp.

108. WIJAYA Miryam Lilian (08/01/96)
Voluntary reciprocity as an informal social insurance mechanism: a game theoretic approach. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 124 pp.

109. VANDAELE Nico (12/02/96)
The impact of lot sizing on queueing delays: multi product, multi machine models. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 243 pp.

110. GIELENS Geert (27/02/96)
Some essays on discrete time target zones and their tails. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 131 pp.

111. GUILLAUME Dominique (20/03/96)
Chaos, randomness and order in the foreign exchange markets. Essays on the modelling of the markets. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 171 pp.

112. DEWIT Gerda (03/06/96)
Essays on export insurance subsidization. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 186 pp.

113. VAN DEN ACKER Carine (08/07/96)
Belief-function theory and its application to the modeling of uncertainty in financial statement auditing. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 147 pp.

114. IMAM Mahmood Osman (31/07/96)
Choice of IPO Flotation Methods in Belgium in an Asymmetric Information Framework and Pricing of IPO's in the Long-Run. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 221 pp.

115. NICAISE Ides (06/09/96)
Poverty and Human Capital. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1996. 209 pp.

116. EYCKMANS Johan (18/09/97)
On the Incentives of Nations to Join International Environmental Agreements. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1997. XV + 348 pp.

117. CRISOLOGO-MENDOZA Lorelei (16/10/97)
Essays on Decision Making in Rural Households: a study of three villages in the Cordillera Region of the Philippines. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1997. 256 pp.

118. DE REYCK Bert (26/01/98)
Scheduling Projects with Generalized Precedence Relations: Exact and Heuristic Procedures. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1998. XXIV + 337 pp.

119. VANDEMAELE Sigrid (30/04/98)
Determinants of Issue Procedure Choice within the Context of the French IPO Market: Analysis within an Asymmetric Information Framework. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1998. 241 pp.

120. VERGAUWEN Filip (30/04/98)
Firm Efficiency and Compensation Schemes for the Management of Innovative Activities and Knowledge Transfers. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1998. VIII + 175 pp.

121. LEEMANS Herlinde (29/05/98)
The Two-Class Two-Server Queueing Model with Nonpreemptive Heterogeneous Priority Structures. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1998. 211 pp.

122. GEYSKENS Inge (4/09/98)
Trust, Satisfaction, and Equity in Marketing Channel Relationships. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1998. 202 pp.

123. SWEENEY John (19/10/98)
Why Hold a Job ? The Labour Market Choice of the Low-Skilled. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1998. 278 pp.

124. GOEDHUYS Micheline (17/03/99)
Industrial Organisation in Developing Countries, Evidence from Côte d'Ivoire. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 251 pp.

125. POELS Geert (16/04/99)
On the Formal Aspects of the Measurement of Object-Oriented Software Specifications. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 507 pp.

126. MAYERES Inge (25/05/99)
The Control of Transport Externalities: A General Equilibrium Analysis. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. XIV + 294 pp.

127. LEMAHIEU Wilfried (5/07/99)
Improved Navigation and Maintenance through an Object-Oriented Approach to Hypermedia Modelling. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 284 pp.

128. VAN PUYENBROECK Tom (8/07/99)
Informational Aspects of Fiscal Federalism. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 192 pp.

129. VAN DEN POEL Dirk (5/08/99)
Response Modeling for Database Marketing Using Binary Classification. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 342 pp.

130. GIELENS Katrijn (27/08/99)
International Entry Decisions in the Retailing Industry: Antecedents and Performance Consequences. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 336 pp.

131. PEETERS Anneleen (16/12/99)
Labour Turnover Costs, Employment and Temporary Work. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 207 pp.

132. VANHOENACKER Jurgen (17/12/99)
Formalizing a Knowledge Management Architecture Meta-Model for Integrated Business Process Management. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 1999. 252 pp.

133. NUNES Paulo (20/03/2000)
Contingent Valuation of the Benefits of Natural Areas and its Warmglow Component. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2000. XXI + 282 pp.

134. VAN DEN CRUYCE Bart (7/04/2000)
Statistische discriminatie van allochtonen op jobmarkten met rigide lonen. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2000. XXIII + 441 pp.

135. REPKINE Alexandre (15/03/2000)
Industrial restructuring in countries of Central and Eastern Europe: Combining branch-, firm- and product-level data for a better understanding of Enterprises' behaviour during transition towards market economy. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2000. VI + 147 pp.

136. AKSOY, Yunus (21/06/2000)
Essays on international price rigidities and exchange rates. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2000. IX + 236 pp.

137. RIYANTO, Yohanes Eko (22/06/2000)
Essays on the internal and external delegation of authority in firms. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2000. VIII + 280 pp.

138. HUYGHEBAERT, Nancy (20/12/2000)
The Capital Structure of Business Start-ups. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2000. VIII + 332 pp.

139. FRANCKX Laurent (22/01/2001)
Ambient Inspections and Commitment in Environmental Enforcement. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. VIII + 286 pp.

140. VANDILLE Guy (16/02/2001)
Essays on the Impact of Income Redistribution on Trade. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. VIII + 176 pp.

141. MARQUERING Wessel (27/04/2001)
Modeling and Forecasting Stock Market Returns and Volatility. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. V + 267 pp.

142. FAGGIO Giulia (07/05/2001)
Labor Market Adjustment and Enterprise Behavior in Transition. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. 150 pp.

143. GOOS Peter (30/05/2001)
The Optimal Design of Blocked and Split-plot experiments. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. X + 224 pp.

144. LABRO Eva (01/06/2001)
Total Cost of Ownership Supplier Selection based on Activity Based Costing and Mathematical Programming. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. 217 pp.

145. VANHOUCKE Mario (07/06/2001)
Exact Algorithms for various Types of Project Scheduling Problems. Nonregular Objectives and time/cost Trade-offs. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. 316 pp.

146. BILSEN Valentijn (28/08/2001)
Entrepreneurship and Private Sector Development in Central European Transition Countries. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. XVI + 188 pp.

147. NIJS Vincent (10/08/2001)
Essays on the dynamic Category-level Impact of Price promotions. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001.

148. CHERCHYE Laurens (24/09/2001)
Topics in Non-parametric Production and Efficiency Analysis. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. VII + 169 pp.

149. VAN DENDER Kurt (15/10/2001)
Aspects of Congestion Pricing for Urban Transport. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. VII + 203 pp.

150. CAPEAU Bart (26/10/2001)
In defence of the excess demand approach to poor peasants' economic behaviour. Theory and Empirics of non-recursive agricultural household modelling. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. XIII + 286 pp.

151. CALTHROP Edward (09/11/2001)
Essays in urban transport economics. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001.

152. VANDER BAUWHEDE Heidi (03/12/2001)
Earnings management in an Non-Anglo-Saxon environment. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2001. 408 pp.

153. DE BACKER Koenraad (22/01/2002)
Multinational firms and industry dynamics in host countries : the case of Belgium. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. VII + 165 pp.

154. BOUWEN Jan (08/02/2002)
Transactive memory in operational workgroups. Concept elaboration and case study. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. 319 pp. + appendix 102 pp.

155. VAN DEN BRANDE Inge (13/03/2002)
The psychological contract between employer and employee : a survey among Flemish employees. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. VIII + 470 pp.

156. VEESTRAETEN Dirk (19/04/2002)
Asset Price Dynamics under Announced Policy Switching. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. 176 pp.

157. PEETERS Marc (16/05/2002)
One Dimensional Cutting and Packing : New Problems and Algorithms. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. IX + 247 pp.

158. SKUDELNY Frauke (21/05/2002)
Essays on The Economic Consequences of the European Monetary Union. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002.

159. DE WEERDT Joachim (07/06/2002)
Social Networks, Transfers and Insurance in Developing countries. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. VI + 129 pp.

160. TACK Lieven (25/06/2002)
Optimal Run Orders in Design of Experiments. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. XXXI + 344 pp.

161. POELMANS Stephan (10/07/2002)
Making Workflow Systems work. An investigation into the Importance of Task-appropriation fit, End-user Support and other Technological Characteristics. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. 237 pp.

162. JANS Raf (26/09/2002)
Capacitated Lot Sizing Problems : New Applications, Formulations and Algorithms. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002.

163. VIAENE Stijn (25/10/2002)
Learning to Detect Fraud from enriched Insurance Claims Data (Context, Theory and Applications). Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. 315 pp.

164. AYALEW Tekabe (08/11/2002)
Inequality and Capital Investment in a Subsistence Economy.Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. V + 148 pp.

165. MUES Christophe (12/11/2002)
On the Use of Decision Tables and Diagrams in Knowledge Modeling and Verification. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. 222 pp.

166. BROCK Ellen (13/03/2003)
The Impact of International Trade on European Labour Markets. K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002.

167. VERMEULEN Frederic (29/11/2002)
Essays on the collective Approach to Household Labour Supply. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. XIV + 203 pp.

168. CLUDTS Stephan (11/12/2002)
Combining participation in decision-making with financial participation : theoretical and empirical perspectives. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2002. XIV + 247 pp.

169. WARZYNSKI Frederic (09/01/2003)
The dynamic effect of competition on price cost margins and innovation. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

170. VERWIMP Philip (14/01/2003)
Development and genocide in Rwanda ; a political economy analysis of peasants and power under the Habyarimana regime. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

171. BIGANO Andrea (25/02/2003)
Environmental regulation of the electricity sector in a European Market Framework. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003. XX + 310 pp.

172. MAES Konstantijn (24/03/2003)
Modeling the Term Structure of Interest Rates Across Countries. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003. V+246 pp.

173. VINAIMONT Tom (26/02/2003)
The performance of One- versus Two-Factor Models of the Term Structure of Interest Rates. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen. 2003.

174. OOGHE Erwin (15/04/2003)
Essays in multi-dimensional social choice. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003. VIII+108 pp.

175. FORRIER Anneleen (25/04/2003)
Temporary employment, employability and training. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

176. CARDINAELS Eddy (28/04/2003)
The role of cost system accuracy in managerial decision making. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003. 144 pp.

177. DE GOEIJ Peter (02/07/2003)
Modeling Time-Varying Volatility and Interest Rates. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003. VII+225 pp.

178. LEUS Roel (19/09/2003)
The generation of stable project plans. Complexity and exact algorithms. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

179. MARINHEIRO Carlos (23/09/2003)
EMU and fiscal stabilisation policy : the case of small countries. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

180. BAESSENS Bart (24/09/2003)
Developing intelligent systems for credit scoring using machine learning techniques. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

181. KOCZY Laszlo (18/09/2003)
Solution concepts and outsider behaviour in coalition formation games. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

182. ALTOMONTE Carlo (25/09/2003)
Essays on Foreign Direct Investment in transition countries : learning from the evidence. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

183. DRIES Liesbeth (10/11/2003)
Transition, Globalisation and Sectoral Restructuring: Theory and Evidence from the Polish Agri-Food Sector. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

184. DEVOOGHT Kurt (18/11/2003)
Essays On Responsibility-Sensitive Egalitarianism and the Measurement of Income Inequality. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

185. DELEERSNYDER Barbara (28/11/2003)
Marketing in Turbulent Times. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

186. ALI Daniel (19/12/2003)
Essays on Household Consumption and Production Decisions under Uncertainty in Rural Ethiopia. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2003.

187. WILLEMS Bert (14/01/2004)
Electricity networks and generation market power. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

188. JANSSENS Gust (30/01/2004)
Advanced Modelling of Conditional Volatility and Correlation in Financial Markets. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

189. THOEN Vincent (19/01/2004)
On the valuation and disclosure practices implemented by venture capital providers. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

190. MARTENS Jurgen (16/02/2004)
A fuzzy set and stochastic system theoretic technique to validate simulation models. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

191. ALTAVILLA Carlo (21/05/2004)
Monetary policy implementation and transmission mechanisms in the Euro area. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

192. DE BRUYNE Karolien (07/06/2004)
Essays in the location of economic activity. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

193. ADEM Jan (25/06/2004)
Mathematical programming approaches for the supervised classification problem. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

194. LEROUGE Davy (08/07/2004)
Predicting Product Preferences : the effect of internal and external cues. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

195. VANDENBROECK Katleen (16/07/2004)
Essays on output growth, social learning and land allocation in agriculture : micro-evidence from Ethiopia and Tanzania. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

196. GRIMALDI Maria (03/09/2004)
The exchange rate, heterogeneity of agents and bounded rationality. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

197. SMEDTS Kristien (26/10/2004)
Financial integration in EMU in the framework of the no-arbitrage theory. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

198. KOEVOETS Wim (12/11/2004)
Essays on Unions, Wages and Employment. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

199. CALLENS Marc (22/11/2004)
Essays on multilevel logistic Regression. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

200. RUGGOO Arvind (13/12/2004)
Two stage designs robust to model uncertainty. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2004.

201. HOORELBEKE Dirk (28/01/2005)
Bootstrap and Pivoting Techniques for Testing Multiple Hypotheses. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

202. ROUSSEAU Sandra (17/02/2005)
Selecting Environmental Policy Instruments in the Presence of Incomplete Compiance. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

203. VAN DER MEULEN Sofie (17/02/2005)
Quality of Financial Statements : Impact of the external auditor and applied accounting standards. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

204. DIMOVA Ralitza (21/02/2005)
Winners and Losers during Structural Reform and Crisis : the Bulgarian Labour Market Perspective. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

205. DARKIEWICZ Grzegorz (28/02/2005)
Value-at-risk in Insurance and Finance : the Comonotonicity Approach. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

206. DE MOOR Lieven (20/05/2005)
The Structure of International Stock Returns : Size, Country and Sector Effects in Capital Asset Pricing. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

207. EVERAERT Greetje (27/06/2005)
Soft Budget Constraints and Trade Policies : The Role of Institutional and External Constraints. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

208. SIMON Steven (06/07/2005)
The Modeling and Valuation of complex Derivatives : The Impact of the Choice of the term structure model. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

209. MOONEN Linda (23/09/2005)
Algorithms for some Graph-Theoretical Optimization Problems. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

210. COUCKE Kristien (21/09/2005)
Firm and industry adjustment under de-industrialisation and globalization of the Belgian economy. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

211. DECAMPS Marc (21/10/2005)
Some actuarial and financial applications of generalized diffusion processes. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

212. KIM Helena (29/11/2005)
Escalation games: an instrument to analyze conflicts. The strategic approach to the bargaining problem. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2005.

213. GERMENJI Etleva (06/01/2006)
Essays on the Economics of Emigration from Albania. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

214. BELIEN Jeroen (18/01/2006)
Exact and Heuristic Methodologies for Scheduling in Hospitals: Problems, Formulations and Algorithms. Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

215. JOOSSENS KRISTEL (10/02/2006)
"Robust discriminant analysis". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

216. VRANKEN LIESBET (13/02/2006)
"Land markets and production efficiency in transition economies". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

217. VANSTEENKISTE ISABEL (22/02/2006)
"Essays on non-linear modelling in international macroeconomics". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

218. WUYTS Gunther (31/03/2006)
"Essays on the liquidity of financial markets". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

219. DE BLANDER Rembert (28/04/2006)
"Essays on endogeneity and parameter heterogeneity in cross-section and panel data". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

220. DE LOECKER Jan (12/05/2006)
"Industry dynamics and productivity". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

221. LEMMENS Aurlie (12/05/2006)
"Advanced classification and time-series methods in marketing". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

222. VERPOORTEN Marijke (22/05/2006)
"Conflict and survival: an analysis of shocks, coping strategies and economic mobility in Rwanda, 1990-2002". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

223. BOSMANS Kristof (26/05/2006)
"Measuring economic inequality and inequality aversion". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

224. BRENKERS Randy (29/05/2006)
"Policy reform in a market with differentiated products: applications from the car market". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

225. BRUYNEEL Sabrina (02/06/2006)
"Self-econtrol depletion: Mechanisms and its effects on consumer behavior". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

226. FAEMS Dries (09/06/2006)
"Collaboration for innovation: Processes of governance and learning". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

227. BRIERS Barbara (28/06/2006)
"Countering the scrooge in each of us: on the marketing of cooperative behavior". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

228. ZANONI Patrizia (04/07/2006)
"Beyond demography: Essays on diversity in organizations". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

229. VAN DEN ABBEELE Alexandra (11/09/2006)
"Management control of interfirm relations: the role of information". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

230. DEWAELHEYNS Nico (18/09/2006)
"Essays on internal capital markets, bankruptcy and bankruptcy reform". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

231. RINALDI Laura (19/09/2006)
"Essays on card payments and household debt". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

232. DUTORDOIR Marie (22/09/2006)
"Determinants and stock price effects of Western European convertible debt offerings: an empirical analysis". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

233. LYKOGIANNI Elissavet (20/09/2006)
"Essays on strategic decisions of multinational enterprises: R&D decentralization, technology transfers and modes of foreign entry". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

234. ZOU Jianglei (03/10/2006)
"Inter-firm ties, plant networks, and multinational firms: essays on FDI and trade by Japanse firms.". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

235. GEYSKENS Kelly (12/10/2006)
"The ironic effects of food temptations on self-control performance". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

236. BRUYNSEELS Liesbeth (17/10/2006)
"Client strategic actions, going-concern audit opinions and audit reporting errors". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

237. KESSELS Roselinde (23/10/2006)
"Optimal designs for the measurement of consumer preferences". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

238. HUTCHINSON John (25/10/2006)
"The size distribution and growth of firms in transition countries". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

239. RENDERS Annelies (26/10/2006)
"Corporate governance in Europe: The relation with accounting standards choice, performance and benefits of control". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

240. DE WINNE Sophie (30/10/2006)
"Exploring terra incognita: human resource management and firm performance in small and medium-sized businesses". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

241. KADITI Eleni (10/11/2006)
"Foreign direct investments in transition economies". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

242. ANDRIES Petra (17/11/2006)
"Technology-based ventures in emerging industries: the quest for a viable business model". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

243. BOUTE Robert (04/12/2006)
"The impact of replenishment rules with endogenous lead times on supply chain performance". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

244. MAES Johan (20/12/2006)
"Corporate entrepreneurship: an integrative analysis of a resource-based model. Evidence from Flemish enterprises". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.

245. GOOSSENS Dries (20/12/2006)
"Exact methods for combinatorial auctions". Leuven, K.U.Leuven, Faculteit Economische en Toegepaste Economische Wetenschappen, 2006.