AUCTION AND BARTER MODELS FOR ELECTRONIC MARKETS

by

Ali Haydar Özer
B.S. in Computer Engineering, Boğaziçi University, 2002
M.S. in Computer Engineering, Boğaziçi University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering
Boğaziçi University
2011

AUCTION AND BARTER MODELS
FOR ELECTRONIC MARKETS

APPROVED BY:

Assoc. Prof. Can Özturan (Thesis Supervisor)	....
Assoc. Prof. Necati Aras	....
Assoc. Prof. Zeki Bayram	....
Assoc. Prof. Haluk Bingöl	....
Prof. Cem Ersoy	....

DATE OF APPROVAL: 16.06.2011

To my

beloved family

ACKNOWLEDGEMENTS

This thesis would not have been possible without the kind support and encouragement of many great people. First and foremost, I would like to express my deepest gratitude to my supervisor Assoc. Prof. Can Özturan, for his continuous support, encouragement, guidance, patience and for especially his kindness during my Ph.D. study. I appreciate his enthusiasm for my research, and also many wonderful ideas and useful comments on this work.

I am grateful to my thesis committee members, Assoc. Prof. Haluk Bingöl and Assoc. Prof. Necati Aras, for the insightful discussions we had and their invaluable feedback on my research over the years.

I am also indebted to Prof. Cem Ersoy and Assoc. Prof. Zeki Bayram, for participating in my thesis jury. I appreciate their comments and suggestions on this thesis.

I would like to thank my dear friends Burak Turhan, Erdinç Saygın, Itır Karaç, Onur Dikmen and all my other friends in our department whom did not leave me alone during these many years. I also owe special thanks to my dear friends for sharing their endless energy with me.

Finally, I would like to thank my beloveds, my dear mother A. Zeynep Özer, my dear father M. Müçteba Özer and my dear brother M. Sadullah Özer for their endless love, support and patience.

ABSTRACT

AUCTION AND BARTER MODELS FOR ELECTRONIC MARKETS

We propose three auction and barter based electronic market models. Our first model is a direct barter model for the course add/drop process in the universities. We model the course add/drop process as a direct barter problem in which add/drop requests can be placed as barter bids. We also introduce a two-level weighting system that enables students to express priorities among their requests while providing fairness among the students. Our second model is the multi-unit differential auction-barter model which augments the double auction model with barter bids so that besides the usual purchase and sale activities, bidders can also carry out direct bartering of items. Our model also provides a mechanism for making or receiving a differential money payment as part of the direct bartering of items, hence, allowing bartering of different valued items. Furthermore, a powerful and flexible bidding language is designed which allows bidders to express their complex preferences of purchase, sell and exchange requests. Our third model is the double auction with limited cover money model. In this model, we propose the use of discrete time double auction institution for the trading of used goods as well as new ones. Our model allows declaration of an amount of cover money so that what is spent on purchased items minus the proceeds of sold items does not exceed this cover money amount. We also introduce a mechanism so that bidders may place multiple item requests in a single bid and limit the maximum number of items to be purchased. We formally define these three models and formulate the corresponding optimization problems. We propose fast polynomial-time network flow based algorithms for optimizing the first and the second models and show that the decision version of the optimization problem for the third model is NP-complete. The performances of our algorithms are also demonstrated on various test cases.

ÖZET

ELEKTRONİK PAZARLAR İÇİN MÜZAYEDE VE TAKAS MODELLERİ

Bu tez içerisinde üç adet müzayede ve takas tabanlı elektronik pazar modeli önerilmektedir. Birinci modelimiz üniversitelerdeki ders ekleme/bırakma süreci için bir doğrudan takas modelidir. Bu modelde öğrenciler ders ekleme/bırakma isteklerini takas teklifi olarak verebilmektedirler. Ayrıca, öğrencilerin isteklerini öncelik sırasına göre belirtebilmelerini ve öğrenciler arasında adaleti sağlayan iki seviyeli bir ağırlıklandırma sistemi de mevcuttur. İkinci modelimiz çok birimli farksal müzayede-takas modelidir. Bu model çift taraflı müzayede sistemine takas özelliği eklemektedir. Bu sayede katılımcılar alış ve satış tekliflerinin yanı sıra takas teklifleri de verebilmektedir. Ayrıca farklı fiyatlı ürünlerin takas edilebilmesini sağlayan, ürünler arasında fiyat farkı belirtme özelliği de sağlanmıştır. Bu özelliklere ilaveten katılımcıların karmaşık satın alma, satış ve takas isteklerini ifade edebilmelerini sağlayan güçlü ve esnek bir teklif dili geliştirilmiştir. Üçüncü modelimiz bütçe sınırlı çift taraflı müzayede modelidir. Bu modelde yeni ve ikinci el ürünlerin ticareti için ayrık zamanlı çift taraflı müzayede sistemi önerilmiştir. Bu model, her bir katılımcının bir bütçe sınırı belirtmesine izin vererek, katılımcının aldığı ve sattığı ürünler arasındaki fiyat farkının belirttiği bütçe sınırı dâhilinde kalmasını sağlar. Ayrıca, bu modelde katılımcıların ilgilendikleri ürünleri tek bir teklifte toplayabilmesini ve almak istediği ürün sayısını sınırlayabilmesini sağlayan bir mekanizma da mevcuttur. Bu üç model ve bu modellere ait eniyileme problemleri matematiksel olarak tanımlanmıştır. Birinci ve ikinci modellerin eniyilemesi için hızlı polinom zamanlı algoritmalar önerilmiş, üçüncü modele ait eniyileme probleminin ise NP-complete sınıfına dâhil olduğu ispatlanmıştır. Önerilen algoritmaların başarımları çeşitli test problemleri üzerinde gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	The ask set
a_{ij}	The requested course
B	The set of bids
b_i	The bid
C	The set of courses in the DALCOM model
c_i	The course
$c_\emptyset$	The null course
D	The set of drop-unless-barter bids
d_i	The course to be dropped
h	The index of the lowest ranked bid
k	The parameter of the k-DA policy
m	The number of courses
$\mathbb{N}$	The set of natural numbers, $\mathbb{N} = \{0, 1, \ldots\}$
p	The reservation price
Q	The set of remaining quotas
q_{c_k}	The remaining quota of the course c_k
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of positive real numbers
R_i	The request set
r_i	The item
s_l	The student
u	The upper limit for the items to be exchanged
w	The weight value of a bid or subbid
x_i	The binary decision variable that denotes whether the bid b_i is satisfied or not
x'_{kt}	The integer variable that denotes the number of units of item to be given by tuple t of the ask set A_k of the bid b_k

x''_{kv}	The integer variable that denotes the number of units of items to be taken by tuple v of the request set R_k of the bid b_k
y_{ij}	The binary decision variable that denotes whether the subbid j of the bid b_i is satisfied or not
$\mathbb{Z}$	The set of integers
$\mathbb{Z}^+$	The set of positive integers
α	The alpha factor in the DALCOM model

LIST OF ACRONYMS/ABBREVIATIONS

CAP	Combinatorial Auction Problem
CT	Course Timetabling
DA	Double Auction
DAB	Differential Auction Barter
DALCOM	Double Auction with Limited Cover Money
DAS	Drop/Add/Swap
FCC	Federal Communications Commission
FCFS	First Come First Served
GPA	Grade Point Average
MIP	Mixed Integer Programming
MUCA	Multi-Unit Combinatorial Auction
MUDAB	Multi-Unit Differential Auction Barter
MUNCA	Multi-Unit Nondiscriminatory Combinatorial Auction
RBS	Registration Bidding System
SCAP	Stable College Admissions Problem
SMR	Simultaneous Multiple-Round
SS	Student Scheduling
UCSP	Used Car Salesman Problem

1. INTRODUCTION

Practically unlimited human needs and wants combined with the limited resources in the world have caused the practice of trading to be an essential social activity throughout the history. This has also led to the natural development of the concept of *market* which is defined as a means for exchanging any type of goods, services and information, i.e. *commodities*, that brings buyers and sellers in contact by providing the necessary regulations and services for trading. Markets provide individuals in a society an ability to exchange scarce resources voluntarily resulting in an increased social welfare. The evidence suggests that markets have been around as long as history and they have also influenced the development of culture [1]. For instance, the earliest known writings found in Mesopotamia are the records of merchants and tax collectors which contain economic information.

Functions provided by a market can be fundamentally categorized into two headings:

(i) *Matching buyers and sellers:* A market brings multiple buyers and sellers to a physical marketplace and matches the buyers and sellers based on the supply and demand profiles. For this matching process, the market provides one or more of the following services [2]: (i) a posting service for sellers that allows them to post their product offerings which are shaped based on the demand information supplied by the market itself, (ii) a search service for buyers which allows them to search and select the product offerings according to their criteria, and (iii) a pricing service for determining transaction prices of the commodities exchanged in the market, for instance the descending price auction is used for pricing in the Dutch flower market [3].

(ii) *Execution and monitoring of transactions:* Aside from the matching service, a market provides the necessary services for facilitating the execution and monitoring of a *transaction* which is the act of exchanging of items of value such as commodities and money between a buyer and a seller. A transaction typically requires the transportation of commodities and transfer of the payments. A market provides necessary information, monitoring and legal services for executing successful transactions among parties.

Recent advances in information technology have made major impact on these functions of markets and provided an ability to shift from the traditional physical markets where the traders meet at a certain place and at a certain time for exchanging commodities, to the electronic markets. Although there is no single definition, in this context an *electronic market* (*e-market*) is defined as an inter-organizational information system that provides one or more of the described functions of a market [4]. Compared to its physical counterpart, an e-market has the following primary benefits:

(i) *Weakens time and space restrictions:* The key feature of an e-market is that it brings multiple buyers and sellers in contact by weakening time and space restrictions [5]. Therefore, an e-market has the potential of attracting more participants than a physical market. For instance, eBay, the world's largest online market, has more than 94 million active users globally [6] and Alibaba.com, the world's biggest business-to-business market has more than one million active users [7]. As the number of participants increases, the higher competition level among the suppliers causes increased supplier innovation [8]. Furthermore, a higher aggregation level of supply and demand is reached which results in increased economic efficiency.

(ii) *Reduces transaction costs:* An e-market can reduce buyers' search costs to obtain information about the product offerings of sellers [4, 9]. The reduced search costs increase the competition level among sellers and prevent them from posting a price which is much higher than the *marginal cost* of a commodity which is the cost of producing one more unit of that commodity. This increases the allocative efficiency of the market, i.e. the efficiency with which a market is allocating resources [10]. Additionally, an e-market can also lower the information sharing and monitoring costs [8] by providing electronic transaction monitoring services. For instance, Garicano et al. [11] analyze the Internet-based firm Autodaq which operates in the wholesale used car auction market of roughly nine million cars per year. They indicate that on the order of 5% decrease in the automobiles' values and 80% decrease in the economic transaction costs are observed.

(iii) *Allows increased product variety:* E-markets allow increased product variety compared to the physical counterparts. For instance, according to the study of Brynjolfsson et al. [12], Amazon and Barnes and Noble have 2.3 million books listed on their online markets whereas a typical physical book store have only 40,000 to 100,000 titles. Similarly, Wal-Mart supercenters which occupy an area of up to 230,000 square-feet have at most one-sixth of the available items in their online version, walmart.com. Brynjolfsson et al. also show that the gain of the consumers from the increased product variety can be up to 10 times higher than the gain obtained from the price reduction in e-markets.

In their seminal article, Malone et al. [13] predict that these benefits of the e-market will cause it to be the dominant institution for coordinating economic activity. However, although there are many positive examples, to date the envisioned utilization level of e-markets has not been reached. Some of the reasons are as follows. First of all, although the ease of accessing information and reduced search costs in e-markets are expected to cause the optimal allocation of resources which also increases the social welfare, in practice they cause the *profit margin*, i.e. the net profit as a percentage of the revenue, of the sellers to approach to zero [5]. This may prevent suppliers from participating in the market and reduce the trading volume greatly as seen in BargainFinder example which was an intermediary CD search service that sorts the CD retailers selling a specific CD based on the sale prices [14]. This caused reduction in the sales of some of the CD retailers, and eventually they blocked this service from accessing their inventory information. Thus, e-markets should provide incentives for both buyers and sellers. A positive example for this is the AUCNET which is a business-to-business type of e-market for used car transactions in Japan [10]. The prices in AUCNET are slightly higher than those in traditional markets, and this attracts more used car suppliers to the market. Increased number of available used cars in the market also attracts the buyers who agree to pay slightly more to avoid the cost of searching and participating in physical markets. Secondly, from buyers' perspective, e-markets are not preferable when the commodity specification is complex and the purchase frequency is relatively low [15]. Furthermore, buyers in especially in business-to-business markets tend to integrate long-term relationships with suppliers in order to benefit from customized products, improved quality and support, and reduced risk of transaction [8, 15, 16]. Finally, it is observed that the increased information flow may also cause information overload and information equivocality which lead to *price dispersion* in e-markets, which is the variation in the prices of the same item across the different sellers of the same item [17].

1.1. Contributions of the Thesis

Considering the aforementioned disadvantages, in order for e-markets to be pervasive, innovative market mechanisms that support more complicated scenarios should be introduced [18]. In this thesis, we propose three different auction and barter based e-market models for three different markets. The proposed models seek to achieve the following objectives:

- The model should be designed specific to the market in which it is intended to be used. It should utilize market-specific features.
- The model should support *multilateral transactions*, i.e. the transactions between two or more participants.
- The model should provide incentives for all participants. It should aim for maximizing the social welfare while providing fair allocation of resources.
- The model should allow participants to express their complex preferences of exchange.
- The model should not bear computational difficulty so that it can be applicable to large e-markets with high number of participants.

Our first model is the direct barter model for the course add/drop process in the universities. Even though course timetabling and student scheduling problems have been studied extensively, not much has been done for the optimization of student add/drop requests after the initial registration period. Add/drop registrations are usually processed with a first come first served policy. This, however, can introduce inefficiencies and dead-locks resulting in add/drop requests that are not satisfied even though they can, in fact, be satisfied. We model the course add/drop process as a direct barter problem in which add/drop requests appear as bids. We formulate the winner determination problem as an integer linear program and show that our problem can be solved polynomially as a minimum cost flow network problem. In our model, we also introduce a two-level weighting system that enables students to express priorities among their requests while providing fairness among the students. We demonstrate both the improvement in the satisfaction of students over the currently used model, and also the fast performance of our algorithms on various test cases based on real-life registration data of our university.

Our second model is the multi-unit differential auction-barter (MUDAB) model which is designed especially for double auction (DA) markets. The MUDAB model augments the well-known DA model with barter bids so that besides the usual purchase and sale activities, bidders can also carry out direct bartering of items. The MUDAB model also provides a mechanism for making or receiving a differential money payment as part of the direct bartering of items, hence, allowing bartering of different valued items. It allows multiple instances of commodities to be exchanged. Furthermore, a powerful and flexible bidding language is designed which allows bidders to express their complex preferences of purchase, sell and exchange requests, and hence increases the allocative efficiency of the market. The winner determination problem of the MUDAB model is formally defined, and a fast polynomial-time network flow based algorithm is proposed for solving the problem. The fast performance of the algorithm is also demonstrated on various test cases containing up to one million bids. Thus, the proposed model can be used in large-scale online auctions without worrying about the running times of the solver.

Finally, the third model proposed in this thesis is the double auction with limited cover money (DALCOM) model. We propose the use of discrete time double auction institution for the trading of used goods and also new ones whereby each bidder can put forward items for sale as well as place bids for purchase. Our double auction market also provides a mechanism for the declaration of an amount of cover money so that what is spent on purchased items minus the proceeds of sold items does not exceed this cover money amount. Since it may be possible for someone to be indifferent to multiple different items or multiple instances of the same item, we introduce a mechanism so that the bidder may place multiple item requests in a single bid and limit the maximum number of items to be purchased. For finding the equilibrium prices, the well known k-double auction policy is used. We formulate the winner determination problem using linear integer programming, and prove that the decision version of this problem is NP-complete and also it is inapproximable unless $P = NP$. We also design a test case generator with wide range of configurable parameters for simulating real markets and prepare a comprehensive test suite. The performance of the CPLEX mixed integer programming (MIP) solver for this problem is demonstrated on this test suite.

In the following chapter, we briefly discuss common auction and barter mechanisms used in e-markets. In Chapter 3, we introduce the direct barter model for course/add drop process. The multi-unit differential auction-barter model and the double auction with limited cover money model are discussed in Chapters 4 and 5, respectively. Finally, the thesis is concluded in Chapter 6.

Parts of this thesis have been published in scientific journals [19–21].

2. A SURVEY OF AUCTION AND BARTER MODELS

There is an enormous literature on the subject of e-markets (see, for example, the surveys [5, 22, 23] and the book [24]) and coverage of the whole literature is beyond the scope of this thesis. We give a brief introduction to common auction and barter models found in the literature in order to provide a basis for clear understanding of the proposed models.

2.1. Auction Models

As defined in [25], an *auction* is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants. Although the precise origin of auctions is not known, the earliest auction found in the historical writings dates back to 500 B.C. [26]. The Greek historian, Herodotus, described Babylonian annual auctions in which young women were sold to men for the purpose of marriage. However in these auctions, instead of starting with low prices and increasing after, the offers for women started from a high price and lowered until a bidder accepted that price. Auctions were also used in the times of the Romans as a system of commercial trade. After a battle, captured prisoners and goods were being auctioned by agents. Although auctions have a history of more than 2500 years, the major development in auction theory has been observed in the past few decades. Especially, advances in information technology have provided us to design more complex and efficient auction models. In this section, we present the most common auction models found in the literature. For a discussion on other auction types and a comprehensive taxonomy, the reader is referred to [27, 28]. Also, an introductory survey on the auction based electronic markets can be found in [29].

2.1.1. Single-Item Auctions

In the traditional *single item auction* model, each item or indivisible bundle of items is auctioned one at a time and the winner is determined simply by picking the highest bidder for each item. Because of its simplicity, this model has been widely used throughout the history and it is still the most popular auction format today.

There are four major single-item auction types [30]:

(i) *The English Auction:* In this well-known open-cry auction type, the auctioneer opens the auction with a *reserve price*, the lowest acceptable price, and proceeds to the higher bids from bidders until no further increase in the offered price occurs. The bidder who offered the highest price gets the item by paying the offered amount. As the term *open-cry* implies, all the bids are known by all the bidders during the auction. In this auction type, the auctioneer has also right to keep the reserve price secret. Because of the high competition level among bidders and inexperienced bidders who bid up the price, the *winner's curse*, that is paying more than the item's utility, is very common. Currently, this is the most widely used auction type around the world.

(ii) *The Dutch Auction:* Although the English auction is the most common auction type, the Dutch auction is the first auction known in the history. In contrast to the English auction, in this auction type, bidding starts at an extreme price and is lowered until a buyer buys the item by calling "mine". The buyer pays the exact price at the time he calls "mine". This auction type has been used in Dutch flower market since the end of the nineteenth century [3].

(iii) *The First-Price Sealed-Bid Auction:* Different from the open-cry auctions, in this auction type, the bids are sealed, and hence are hidden from the other bidders. Generally, each bidder is allowed to submit only one bid which makes the preparation of a bid extremely important. There are two phases of this auction type. A bidding phase in which bidders submit their bids and a resolution phase in which the winner of the auction is determined. As in the English auction, the winner pays the exact amount he offered in his bid. Sealed first-price auctions are commonly used in government tenders.

(iv) *The Second-Price Sealed-Bid Auction:* This auction type is also known as the *Vickrey auction* which has been introduced by Vickrey et al. [30]. As the first-price sealed-bid auction, the bids are sealed. However, in this auction type, the winner of the auction pays the amount of the second highest bid instead of the amount of his own bid. Although this auction format is rarely used, it is important theoretically since it provides bidders an incentive to offer their true utility values in their bids.

These types of auctions can also be extended to multi-unit case in which more than one identical or equivalent units of an item are auctioned. For detailed discussion on multi-unit single-item auctions see [31].

2.1.2. Combinatorial Auctions

Single-item auctions are suitable when the value of each item is unrelated to the values of other items for every bidder. However, there may be complementarities and substitutabilities between items [32]. Assume that in an electronic equipment auction, several different brands of televisions and video recorders are to be auctioned. The valuation of a bidder on the bundle of a television and a video recorder can be higher than the sum of the valuations of the television and the video recorder alone. In this case, there is complementarity between the television and the video recorder for this bidder. Conversely, the valuation of a bidder on the bundle of two different brands of televisions can be lower than the sum of the separate valuations. In this case, there is substitutability between televisions of different brands for this bidder. Formally, *complementarity* between items i and j exists if $g_b(\{i, j\}) > g_b(\{i\}) + g_b(\{j\})$ where $g_b(S)$ is the gain of getting a set of items S for a bidder b. *Substitutability* between items i and j exists if $g_b(\{i, j\}) < g_b(\{i\}) + g_b(\{j\})$.

If there are complementarities between different items, single-item auctions may provide inefficient allocation. Although *parallel auction* model in which items are auctioned simultaneously can increase the allocative efficiency by reducing uncertainty, this does not help us in solving the *complementarity problem* that involves complementarities between items [33]. *Combinatorial auction* model solves this problem by allowing package bidding, i.e. bidding on combinations of different items [34, 35]. In this model, all items are available to bidders and bidders are free to express their own valuations of any combination of items. The combinatorial auction model is applicable to many real-world situations such as auctions for radio spectrum rights [36], airport slot allocations [37], transportation services [38–40], course registrations [41], and commercial time slot allocations [42]. Among these, probably the most famous auctions are the Federal Communications Commission (FCC) electromagnetic spectrum auctions [43–46]. Since 1994, FCC has conducted more than 50 auctions in the type of simultaneous multiple-round (SMR) auctions. In SMR, licenses that have complementarities are available for bidding in parallel. Auctions are conducted in successive rounds lengths of which are announced by FCC. After each round, results are processed and made public. Until the next round, bidders go over their bid strategies and adjust their bids if necessary. Auction ends when no new bid is submitted during a round. Although package bidding is not allowed in the original SMR auctions, the laboratory experiments have demonstrated that allowing package bidding will improve the spectrum auction outcomes in the presence of complementarities [47].

In combinatorial auction model, it is possible to solve the *substitutability* problem by introducing one or more dummy items for each substitutable item [32]. The role of dummy items is to allow bidders to express exclusive-or relationship between bids. For instance, if a bidder wants to get one television of brand A or B, one dummy item should be introduced and two separate bids, one bid for combination of dummy item and television of brand A and another bid for combination of dummy item and television of brand B, must be submitted by the bidder. Although this trick helps solving substitutability problem, the number of bids increases combinatorially with the number of substitutable items.

Combinatorial auction model provides economically better allocation of items at the expense of computational difficulty. The *winner determination problem of combinatorial auction model* (CAP) as formulated by Rothkopf et al. [48] and Sandholm [33] is an instance of weighted set packing (and also weighted independent set) problem which is known to be NP-hard [48–50]. For unrestricted CAP, many optimum search algorithms have been proposed [32, 33, 51–54]. It must be noted that although specialized algorithms perform better in some test cases compared to commercial general purpose MIP software such as CPLEX [55], in other cases they fall behind [51, 53, 54]. Therefore, general purpose MIP solvers can be considered as a good choice among the optimum solvers for CAP. For detailed discussion about optimum search techniques see [35, chap. 14].

The single unit combinatorial auction model provides economically efficient allocations when the bidders are interested in bundles of items. However, they are inappropriate for situations where multiple instances of items are auctioned. Since a bidder would not be interested in a specific unit, he should place a separate bid for each combination of items he wants to buy. For instance, if the bidder wants to get 50 keyboards and 60 mice out of 200 keyboards and 300 mice in a single unit combinatorial auction, he must bid $\binom{200}{50}\binom{300}{60}$ times where $\binom{n}{r}$ is combination of r out of n. Multi-unit combinatorial auction (MUCA) model solves this problem by representing identical items as multiple units of single item and allowing bidders to bid on instances of items [56–59]. For further discussion on combinatorial auctions, the reader is referred to [34].

If multiple instances of items are to be auctioned, the MUCA model provides efficient bid representation. Although substitutability problem between identical units of items is solved with this model, substitutability between different items is not considered. If the bidder does not differentiate between two or more different items, the MUCA model becomes insufficient for representing such preferences. As an example, assume that 100 units of each of the four different types of a keyboard (e.g. type A,B,C and D) are to be auctioned in the MUCA model. Consider the following two scenarios:

- A bidder wants to buy 100 keyboards of type A or C. In order to express this preference, one dummy item must be introduced and $\binom{2+100-1}{100} = 101$ bids should be placed.
- Likewise, if a bidder wants to buy 100 keyboards without differentiating any four keyboard types, one dummy item must be introduced and $\binom{4+100-1}{100} = 176,851$ bids should be placed.

In order to overcome this inefficiency in preference expression, Özer et al. [19] have proposed a new auction model called *multi-unit nondiscriminatory combinatorial auction* (MUNCA) model. In this model, bidders may encode their preferences of substitutability to bids easily by declaring a list of nondiscriminatory items. In the above example, MUNCA model would enable the bidder to express his preferences by placing only one bid in each of the above scenarios. For further discussion on this model, the reader is referred to [19, 60].

2.1.3. Reverse Auctions

In the aforementioned single-item and combinatorial auction models, a single seller posts one or more items for sale and multiple buyers submit bids for purchasing these items. In the *reverse auction models*, the roles of buyers and sellers are reversed. A single buyer holds a reverse auction for the procurement of one or more items and multiple sellers submit asks for selling the requested items. In this case, the objective of the auction is to lower the prices of items for the benefit of the buyer.

As in the combinatorial auctions, if there are complementarities between different items, *combinatorial reverse auction* model can be used. In combinatorial reverse auctions, buyers can submit asks for bundles of items. However, as in the case of the combinatorial auction model, the winner determination problem is NP-hard [58].

2.1.4. Double Auctions

In *double auction* (*DA*) model, multiple sellers and buyers coexist in the auction and can submit multiple *asks* (sale bids) and *bids* (purchase bids) simultaneously for well-defined commodities. Beginning with Smith [61], many laboratory experiments with DA rules have been carried out (see [62] and [63] for a survey). The experiments demonstrate that the DA institution results in high allocative efficiency even with a small number of traders [64–66] which also explains the wide usage of the DA institution in the exchanges. The popularity of the institution has motivated researchers to propose and study variants of the DA. The *single-unit DA* is the base model in which each commodity in the market is considered as a unique item. The *multi-unit DA* [67, 68] is an extension to the single unit DA which allows multiple instances of a commodity to be traded in the auction. Kalagnanam et al. [69] introduce a multi-unit DA institution with assignment constraints and propose a network flow algorithm for finding the optimum allocation of commodities. In the *(multi-unit) combinatorial DA* [70, 71], the institution further allows package (combinatorial) bidding in which the participants can submit bids on bundles of items instead of a single item as in combinatorial auctions.

According to the market clearing interval, there are two types of the DA model. In the *continuous time* DA institution, asks and bids can be submitted and retracted any time during the auction, and exchanges of items are carried continuously. Depending on the active set of asks and bids, a public list of best orders, asks with lowest price and bids with highest price, is maintained. If a seller offers a price that is smaller than or equal to the bid with the highest price or if a buyer offers a price that is greater than or equal to the ask with the lowest price, the corresponding ask and bid pair are matched and removed from the system. The continuous time DA is the preferred auction format in the stock exchanges.

In the *discrete time* DA institution, which is also known as the *call-market* [72] or the *clearinghouse* [73], traders submit sealed asks and bids during a predefined trading period. At the end of this period, the asks and bids are sorted according to their prices, and supply/demand profiles for the commodities, i.e. the supplied and demanded quantities at each price, are generated. Using these profiles, market equilibrium prices are found. The market is then cleared by matching the lowest priced ask to the highest priced bid and continuing until the prices cross the equilibrium price threshold. This market clearing procedure is computationally easy [28, 74].

For determining the market equilibrium prices in a discrete time DA institution, k-DA policy can be used [75]. In two player k-DA policy, first a parameter $k \in [0, 1]$ is determined, and then the equilibrium price is defined as:

$$k \cdot b + (1 - k) \cdot a$$

where a and b are the prices of the matched ask and bid, respectively ($a \leq b$). The parameter k defines the distribution ratio of the utility $(b - a)$ between the seller and the buyer. For instance, in 1/2-DA, the utility is equally shared. The border cases $k \in \{0, 1\}$ have strategic importance. If $k = 0$, the seller determines the equilibrium price. This policy is called *seller's offer* DA. Conversely, if $k = 1$, then the policy is called *buyer's bid* DA in which the buyer determines the equilibrium price. The k-DA policy can simply be extended to the multilateral case where there are multiple buyers and sellers of the same item [76].

In terms of allocative efficiency, the expected efficiency of a discrete time DA institution is higher than that of the continuous time DA [77, 78]. Despite this inefficiency, the continuous time DA is still the preferred institution in many financial markets since it is believed that the high market throughput of the continuous time DA institution compensates the efficiency loss [79].

For further details on DA institution, the reader is referred to the surveys [73, 78].

2.2. Barter Models

2.2.1. Direct Bartering

Bartering, or *direct bartering* in this context, is defined as trading by exchange of commodities rather than by the use of any medium of exchange such as money. Direct bartering is the first known trade format and has been used throughout the history especially till the invention of the commodity money. In 1888, Jevons [80] stated the difficulty of finding "double coincidence of wants", that is finding two individuals who want to exchange their items. This difficulty is the major weakness of the direct bartering that severely restricted its usage in the exchanges. However, the recent advances in information technology have enabled us to utilize multilateral direct bartering, i.e. bartering involving two or more parties, which makes the bartering institution suitable for many specialized markets, such as trading of computational resources, internet domain names and electronic media over the barter grids [81, 82].

As in the auction models, depending on whether multiple units of an item can exist in the market and whether the combinatorial exchange is allowed or not, direct bartering models can be classified into four categories [81]:

(i) *Single-instance single-item*: In this model, each item in the market is considered as unique and only one-to-one exchanges are allowed that is each item may be bartered for another single item. For instance, a bidder may place a barter bid stating that he wants to barter his own item A for another item C. The associated optimization problem aims to maximize the number of bartered items by extracting a maximum set of arbitrary length distinct barter cycles. An example for a barter cycle of length three is

$$\text{Bidder 1} \xrightarrow{\text{gives item } A \text{ to}} \text{Bidder 2} \xrightarrow{\text{gives item } B \text{ to}} \text{Bidder 3} \xrightarrow{\text{gives item } C \text{ to}} \text{Bidder 1}$$

The barter requests (bids) in this model can be represented using a directed graph. In this graph, each item in the market is represented with a vertex and each barter bid is represented with an arc such that the arc originates from the vertex representing the owned item and is directed towards the vertex representing the item to be exchanged for. Using this representation, the optimization problem can be reduced to the *maximum vertex disjoint problem* which can also be reduced to the *assignment problem* [83, 84]. This problem is solvable in $\mathrm{O}(n^3)$ time where n is the number of barter requests. This model corresponds to the single-unit single-item auction model in the auction taxonomy.

(ii) *Multiple-instance single-item*: This model extends the single-instance model such that there can be multiple units of each item in the market. However, the allowed exchange type is still one-to-one such that a single unit of an item can be bartered for single unit of another item. For instance, a bidder can offer to barter his own three units of item A with three units of another item B. Ozturan [82] solves the optimization problem of this model by reducing it to the *minimum cost network flow problem* in strongly polynomial time. This model corresponds to the multi-unit single-item auction model.

(iii) *Single-instance multiple-item*: As in the single-instance single-item model each item is considered as unique, and so there can only be single instance of each item in the market. However, the model now allows many-to-many exchanges, that is a bundle of any number of items of a participant can be bartered for another bundle of any number of items in the market. For instance, a barter request in a form such as barter a bundle of item A and item B for item C is allowed. The optimization problem for this model is NP-Hard [81]. This model corresponds to the single-unit combinatorial auction model.

(iv) *Multiple-instance multiple-item*: This model extends the single-instance multiple-item model to support multiple units of items. Thus, there can be multiple instances of each item and bids on the combinations of items are allowed. An example barter request is barter a bundle of two units of item A and three units of item B for a bundle of one unit of item C and two units of item D. As in the single-instance case, the associated optimization problem is NP-Hard [81]. This model corresponds to the multi-unit combinatorial auction model.

Ozturan [81, 82] provides the mathematical description of these direct barter models and the solution procedures. López et al. [85] have also studied electronic barter architectures that utilize direct bartering mechanism and present a formal framework for such architectures using process algebra. They extend their framework in [86] so that the transaction and shipping costs are included.

Bartering has diverse application areas. For instance, Buttyán et al. [87] propose a direct bartering mechanism to increase the message delivery rate in opportunistic networks in which the messages are transferred through the intermediate agents in a store-carry-and-forward manner. In the proposed mechanism, a mobile agent can only get a message from another agent if it uploads another message in return. This prevents agents from being selfish and refusing to distribute messages for the benefit of other agents in the network, and increases the message delivery rate.

Another example is BitTorrent [88] which is a peer-to-peer file distribution system. It uses a direct bartering mechanism to provide incentive to users for robust distribution of files. The users who want to download the same file can barter the downloaded parts of the file among themselves in order to prevent a possible bandwidth bottleneck which could be observed if all users download from a single source. However, the direct bartering mechanism used in BitTorrent is bilateral. Multilateral bartering in the form of reciprocal trade has also been used in other peer-to-peer file distribution systems such eMule [89] and PACE [90]. Bilateral and multilateral approaches in peer-to-peer systems are compared in [91].

Direct bartering is also used in kidney exchanges [92–95]. Kidney failure is a common disease which unfortunately has no permanent treatment except a kidney transplant. Since the demand for transplants cannot be met with the deceased-donor kidneys, patients also look for a living donor who can give one of his two kidneys. However, for a kidney transplant to be made the donor and the patient must be blood-type and tissue-type compatible. Thus, finding a compatible living donor is more difficult than just finding a living donor. A solution for this compatibility problem is kidney exchanges. In a kidney exchange, the living donors of the patients are matched using a direct barter mechanism. Consider the following example. Assume that Donor D_a wants to donate his kidney to Patient P_a but they are incompatible. Similarly, Donor D_b wants to donate his kidney to Patient P_b but they are incompatible, too. However, it may be the case that Donor D_a is compatible with Patient P_b and Donor D_b is compatible with Patient P_a. Then, using the direct barter mechanism, Donor D_a can be matched with Patient P_b and Donor D_b can be matched with Patient P_a. This direct bartering mechanism corresponds to single-instance single-item model with one major difference. In single-instance single-item model, the length of the barter cycles can be arbitrary. For kidney exchanges on the other hand, because of the necessity of doing transplants simultaneously, the length of cycles should be limited. This makes the resulting optimization problem NP-Hard. However, for this problem, Abraham et al. [95] propose efficient optimum algorithms that can handle 10,000 patients, and thus, allow the nation-wide exchanges to be established.

2.2.2. Reciprocal Trade

The difficulty of finding "double coincidence of wants" for the direct bartering can be overcome by the reciprocal trade [96, 97]. In reciprocal trade, also known as barter trade, individuals that decide to participate in barter become customers of a trade exchange. Each customer has an account of barter units controlled by the exchange. Different from the direct bartering, trading of goods or services are made in multilateral fashion using barter units. When a customer sells a good, the amount of the good in terms of barter units is deposited to his account. Similarly, when he buys a good, the amount of the good is withdrawn from his account. Allowing trading of goods of unequal value and overcoming the necessity of finding double coincidence of wants, makes barter trade a more preferred alternative than the direct bartering in commodity exchanges.

3. A DIRECT BARTER MODEL FOR THE COURSE ADD/DROP PROCESS

3.1. Introduction

In universities, course timetabling (CT), student scheduling (SS) and add/drop processes involve the coordination of various resources and entities. CT basically deals with the allocation of time slots and classrooms to courses by taking into consideration issues such as preferences of instructors and classroom locations. Given a timetable, in SS phase, students select courses according to their needs and preferences. Because of course and section quota restrictions or enrollment balancing requirements among the sections, it is not possible to satisfy the needs and preferences of all the students. Therefore, some policy or algorithm needs to be employed in SS phase for the assignment of students to courses and sections. During the add/drop phase, a readjustment of the assignment solution in SS phase basically takes place by the addition, dropping and swapping of courses and/or sections. In the literature, phases CT and SS have been extensively studied (see, for example, surveys [98–101]). Some approaches tackled either CT or SS exclusively. Some approaches coupled these two phases and solved the combined course timetabling and student scheduling problem. In this chapter, our focus will be on the add/drop process. Not much has been done for this phase - we are aware of only one work (that of Graves et al.'s [41]) that addresses the add/drop process. The add/drop process has an important difference from that of CT and SS. A student may have been already assigned to a seat in a course or section from SS phase and he may want to swap (barter) this seat that he owns with another seat owned by other students in another course or section. Hence, one can say that whereas CT and SS phases can be modeled as an assignment problem, for the add/drop process bartering is a more appropriate model.

We were motivated to develop a direct barter model for the add/drop process because of some problems we noticed during add/drop periods at our Boğaziçi University. Since 1998, a web based online registration system has been used for course registration [102]. Before the beginning of each semester, students are admitted to the system and are allowed to take courses if both prerequisites of the courses are satisfied and the quotas of the courses permit. The system works on a first come first served (FCFS) policy basis and at the beginning of each registration period, a race occurs among students for popular courses. Generally, the quotas of the popular courses are filled within the first few hours of online registration period. After the registration period, the semester begins and during the first week of the semester, students attend and evaluate their courses. At the end of this week, add/drop period of one week begins and the students are allowed to change their courses and/or sections of their courses. Because of the FCFS basis of the system and the quota restrictions, when a student drops a course, he may not be able to take it again. This situation forces a student who wants to change his course, first try to add a new course, and then drop the old course. Although this does not pose a problem if the quotas of the courses are not full, it does for the popular courses. It is observed in Boğaziçi University student registration system that the current FCFS based system causes deadlock situations, and hence reduces the total satisfaction of students. Although different implementations of FCFS approach exist in different registration systems, all FCFS based systems are prone to the same problem. For instance, in UniTime [103, 104], which is an open-source enterprise system for automated construction of course timetables and student schedules, when a student wants to add a course which is not available, the student is assigned to the wait-list of that course. Wait-lists are processed automatically in FCFS manner as one seat becomes available for the corresponding course. Therefore, since a student who wants to change his course cannot be sure whether he would be assigned to the new course, he would not want to drop the course he has already assigned until he obtains a seat in the new course. Thus, this would also lead to the same problem.

In order to increase the efficiency of add/drop process in terms of students' satisfaction compared to the current FCFS based system of our university, a direct barter model for the course add/drop process is proposed. The objective of the model is to increase the total satisfaction of students while preserving fairness among them. For this purpose, along with the usual add and drop requests, this model allows students to barter the courses they want to drop for the courses they want to add. Students express their requests through submitting multiple add, drop and barter bids and in each add and barter bid, they can declare a set of alternative courses to be added. Besides, in this model, they can indicate relative priorities of their bids and the courses they want to register for. For instance, if a student prefers course A over course B, and course B over course C, he just declares $A \succ B \succ C$. Furthermore, students can request the same course or the same set of courses in multiple bids and can also declare restriction sets in which only one course can be added to their schedule.

In this chapter, we contribute a formal development of the model. We present a network flow based algorithm that allows us to solve the problems in strongly polynomial time. We also compare the solutions of our model with that of the FCFS approach based on real-world student registration data and present the performance of our algorithms on various tests.

In the next section, we present an example with which we explain our model for the course add/drop process. In Section 3.3, we formally define and formulate our model using integer programming. Then, in Section 3.4, we present a minimum cost network flow solution of our problem and in Section 3.5, we present the experimental results. A review of the related literature is given in Section 3.6. Finally, the chapter is concluded in Section 3.7.

3.2. A Motivational Example and the Model

In this section, we present an example scenario for the add/drop process on which we explain our direct barter model. Assume that during the registration period, students Ali, Mehmet, Ayşe and Aslı have been registered for courses STS 401.01, SOC 101.01, ESC 301.01 and SOC 101.01, respectively. Murat, on the other hand, has been registered for both STS 401.02 and PSY 101.01. Suppose that during the add/drop period, the students declare *add*, *drop*, and *barter bids* as shown in Table 3.1.

Table 3.1. Example problem for illustrating add, drop, and barter bids.

```
Bids:
1.  Ali     : STS 401.01 → {PSY 101.01, STS 401.02}
2.  Mehmet  : SOC 101.01 → {STS 401.01}
3.  Ayşe    : ESC 301.01 → {SOC 101.01}
4.  Murat   : STS 401.02 → {SOC 101.01, ESC 301.01}
5.  Murat   : PSY 101.01 →* {ESC 301.01}
6.  Elif    :          ∅ → {STS 401.01, STS 401.02}
7.  Elif    :          ∅ → {SOC 101.01}
8.  Aslı    : SOC 101.01 →  ∅

Remaining Quota Information:
• STS 401.02 : 1 student
```

Bids 1-5 are examples of a barter bid. In a *barter bid*, the left hand side of the arrow indicates the course to be dropped and the right hand side indicates the course to be added. A barter bid as the name suggests enforces the student to drop the course on the left hand side if he adds the new course on the right hand side. For instance, in bid 3 Ayşe wants to drop ESC 301.01 if she could add SOC 101.01 to her course list. Bids 6 and 7 are examples of an add bid. An *add bid* states that the student wants to add the course on the right hand side without dropping any other course. Likewise, a *drop bid*, e.g. bid 8, states that the student wants to drop the course on the left hand side without adding any other course.

Bids 1, 4 and 6 are different from the others in terms of having a *request set* of more than one course on the right hand side. These bids are called *multi-bids.* A *multi-barter bid* states that the student is indifferent, at least to some degree, to the set of requested courses and he is willing to drop the course on the left hand side if he could add *any one* of the courses in this set. Similarly, a *multi-add bid* states that the student wants to add any one of the courses in this set without dropping any other course. Since drop bids are not restricted with quota constraints, they should always be satisfied. Therefore, we do not need to explicitly incorporate multi-drop bids in the model.

Multi-bids can be considered as combinations of two or more single bids which are XOR'ed. For instance, in bid 1, Ali wants to add either PSY 101.01 or STS 401.02 but not both and drop STS 401.01 on the condition that his add request is satisfied. This bid can be represented as a combination of two XOR'ed bids, STS 401.01 $\rightarrow$ {PSY 101.01} and STS 401.01 $\rightarrow$ {STS 401.02}. Bid 1 is satisfied if exactly one of these bids is satisfied. By introducing multi-barter bids, without losing generality, we can now safely assume that there are no two barter bids of a student that have the same course on the left hand side since such bids can be combined and represented as one multi-barter bid. In addition to the multi-barter bid mechanism, the model allows a student to mark a barter bid (either single or multi-barter) as *drop-unless-barter* meaning that the student wants to barter the course on the left hand side for a course in the request set given on the right hand side. However, if bartering is not possible, then he wants to drop the course on the left hand side. In the given example, it is indicated using a star above the arrow of the bid 5. In this bid, Murat wants to barter PSY 101.01 for ESC 301.01 and if ESC 301.01 cannot be added, Murat wants to drop PSY 101.01. Again, by further introducing drop-unless-barter mechanism, drop and barter requests can be combined and again without losing generality, we can state that there cannot be *any two bids* of a student that have the same course on the left hand side.

The add/drop process based on the direct barter model is a batch process and consists of two phases, a *bid submission phase* in which the students are allowed to submit bids to the system or retract bids from the system and a *solution phase* in which the optimum solution is calculated. Depending on the duration of the add/drop period, these phases can be repeated as many times as necessary. For instance, for each day of the add/drop period, the bids can be collected from the students throughout the day and the solution can be calculated at the end of the day and announced to the students afterwards.

3.2.1. Expressing Preferences - Weighted Model

Although the described unweighted model helps to increase the total satisfaction of students, it can further be improved by assigning weights to the bids. In the weighted model, add, drop, and barter bids are defined respectively as follows:

$$
\begin{aligned}
&c_{\emptyset} \xrightarrow{w_i} \{(a_{i1}, w_{i1}), (a_{i2}, w_{i2}), \ldots, (a_{ip}, w_{ip})\} \\
&d_i \xrightarrow{w_i} \{(c_{\emptyset}, 0)\} \\
&d_i \xrightarrow{w_i} \{(a_{i1}, w_{i1}), (a_{i2}, w_{i2}), \ldots, (a_{ip}, w_{ip})\}
\end{aligned}
$$

where i is the index of the bid, w_i is the weight of the bid i, and d_i is the course to be dropped. $c_{\emptyset}$ denotes the *null course* used for representing the course to be dropped or added for add and drop bids respectively. In the weighted model, weights are assigned not only to the bids, but also to the requested courses. Therefore, the request set contains tuples (a_{ij}, w_{ij}) where a_{ij} is the requested course, w_{ij} is the associated weight, and p is the number of the requested courses.

By assigning weights to the bids and the requested courses, the model becomes more powerful in the sense that it enables students to express their preferences inside the bids. For instance, weight values can be assigned to a student's bids indicating the degree of his preferences for his bids. The weight of the most favored bid would be the highest and the least favored would be the lowest. Likewise, for each multi-bid, weight values can also be assigned to the requested courses in the request set if he is not totally indifferent to these courses. Considering the quota restrictions and the submitted bids, among the possible courses in the request set, the one with the highest weight would be added to the student's course list. Besides the ability to express preferences among the bids and the requested courses, the weighted model also enables favoring some students over others. Special students such as graduating students can also be favored officially by their department by increasing or maximizing the weights of their bids. This will ensure that if the quotas of the courses are available then these students will be the first to add the courses they want. Similarly, using the same mechanism, successful students, i.e. the students with higher grade point average (GPA), can also be favored depending on the policy of the university.

3.3. Formulation of the Model

The weighted direct barter model is formally defined as follows: let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of m courses and $Q = (q_{c_1}, q_{c_2}, \ldots, q_{c_m})$ be the tuple of remaining quotas where q_{c_k} is the remaining quota of course c_k ($1 \leq k \leq m$, $q_{c_k} \in \mathbb{N}$). Let $S = \{s_1, s_2, \ldots, s_t\}$ be the set of t students. We define B_l as the set of bids submitted by a student s_l and the set of all bids, B, is defined as $B = \bigcup_{l=1}^{t} B_l$. Each bid is denoted by a triplet, $b_i = (d_i, w_i, R_i)$, where d_i is the course to be dropped for *barter* and *drop* bids or the null course, $c_\emptyset$, for *add* bids ($d_i \in C \cup \{c_\emptyset\}$), $w_i \in \mathbb{R}^+$ is the weight and R_i is the *request set* of the bid b_i. The request set of a bid is either $\{(c_\emptyset, 0)\}$ for *drop* bids or a set of two tuples, $R_i = \{(a_{i1}, w_{i1}), (a_{i2}, w_{i2}), \ldots, (a_{ip}, w_{ip})\}$, for *barter* and *add* bids. Each tuple (a_{ij}, w_{ij}) in R_i indicates the requested course, that is the course to be added, and the associated weight respectively ($a_{ij} \in C, w_{ij} \in \mathbb{R}$). Finally, the set $D \subseteq B$ denotes the bids which are marked as *drop-unless-barter*.

A bid b_i is called *satisfiable* if at least one of the courses in the request set has one or more remaining quota or there exists at least one satisfiable bid whose course to be dropped is in the request set of b_i. Formally, given $b_i = (d_i, w_i, R_i)$ and $b_l = (d_l, w_l, R_l)$ the following predicate is true: $\forall i(\text{satisfiable}\,(b_i) \Leftrightarrow \exists j\,((a_{ij}, w_{ij}) \in R_i \wedge ((q_{a_{ij}} \in Q \wedge q_{a_{ij}} > 0) \vee \exists l\,(satisfiable\,(b_l) \wedge d_l = a_{ij}))))$

By the definition, all drop bids are satisfiable. The objective of the model is to find the set of satisfiable bids that maximizes the sum of the weights, that is the sum of both bid weights and weights of the requested courses, and hence the *total satisfaction* of students.

In order to formulate the model using integer programming, two binary variables are introduced. The binary decision variable x determines the satisfied bids and y determines the requested course which is added if the corresponding bid is satisfied.

Formally,

$$x_i = \begin{cases} 1, & \text{if bid } i \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad y_{ij} = \begin{cases} 1, & \text{if course } a_{ij} \text{ is added for bid } i \\ 0, & \text{otherwise} \end{cases}$$

It should be noted that for a drop-unless-barter bid b_i, the meaning of satisfaction is slightly different. $x_i = 0$ means that the student drops the course without adding any other course and $x_i = 1$ means that the student barters the course for another course. The integer programming formulation of the model is as follows:

$$\text{maximize} \sum_{\forall i \,|\, b_i \in B} (w_i\, x_i + \alpha \cdot \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i} w_{ij}\, y_{ij}) \tag{3.1}$$

subject to

$$\sum_{\forall i \,|\, b_i \in (B \backslash D) \wedge d_i = c_k} x_i \; - \sum_{\forall i,j \,|\, b_i \in B \wedge (a_{ij}, w_{ij}) \in R_i \wedge a_{ij} = c_k} y_{ij} \;\geq\; -\left(q_{c_k} + \sum_{\forall i \,|\, b_i \in D \wedge d_i = c_k} 1 \right) \quad (\forall k \,|\, c_k \in C) \tag{3.2}$$

$$x_i \; - \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i} y_{ij} \;=\; 0 \quad (\forall i \,|\, b_i \in B \wedge R_i \neq \{(c_\emptyset, 0)\}) \tag{3.3}$$

$$x_i \;=\; 1 \quad (\forall i \,|\, b_i \in B \wedge R_i = \{(c_\emptyset, 0)\}) \tag{3.4}$$

$$\sum_{\forall i,j \,|\, b_i \in B_l \wedge (a_{ij}, w_{ij}) \in R_i \wedge a_{ij} = c_k} y_{ij} \;\leq\; 1 \quad (\forall k, l \,|\, c_k \in C \wedge s_l \in S) \tag{3.5}$$

$$x_i, y_{ij} \;\in\; \{0, 1\} \quad (\forall i, j) \tag{3.6}$$

Note that in this formulation, the objective line in Eq.(3.1) maximizes the sum of the bid weights and the weights of the courses in the request sets. In this equation, α factor is a constant positive number to be determined according to the actual weight values and the number of bids which is described in the next section. Eq.(3.2) enforces quota restrictions of the courses. For each course, the number of students dropping the course (including the drop and drop-unless-barter bids) plus the remaining quota of the course should be greater than or equal to the number of students who added the course. Eq.(3.3) expresses the satisfaction criterion: an add bid or a barter bid is satisfied if exactly one of the courses in the request set is added. Eq.(3.4) ensures that all the drop bids are satisfied. Finally, Eq.(3.5) prevents students from adding the same course to their schedule more than once.

3.3.1. Determining the Weight Values and the α Factor

The main objective of the direct barter model is to increase the total satisfaction of students while preserving fairness among them. In this section, we propose a method for defining the parameters of the model, that are the weights of the bids, w_i, the weights of the requested courses, w_{ij}, and the α factor, in accordance with this objective.

In this method, each student, s_l, is responsible for ranking his bids according to his preference instead of defining the actual weights of the bids. Based on this ranking, he constructs his *preference list*, a permutation of his bids sorted in descending order of his preference. Then, this preference list is used as B_l, the set of bids submitted by the student s_l. So, in the set $B_l = \{b_l^{(1)}, b_l^{(2)}, \ldots, b_l^{(u)}\}$, the bid $b_l^{(1)}$ is the most preferred bid with highest rank number of 1 and the bid $b_l^{(u)}$ is the least preferred bid with the lowest rank number of u (i.e. $\forall l \,:\, b_l^{(1)} \succ b_l^{(2)} \succ \ldots \succ b_l^{(u)})$. Note that, for this definition we use a different indexing scheme for referring the bids in the set B_l in order to prevent confusion with the indexing scheme for referring the bids in the bid set B. Subscript l of a bid $b_l^{(r)}$ denotes the owner of the bid and the superscript r, which is the index number in the set B_l, denotes the rank of the bid. We also denote the weight of a bid $b_l^{(r)}$ with $w_l^{(r)}$. Given the ordered set of bids B_l of the student s_l, the weight of a bid $b_l^{(r)}$ is defined as follows:

$$w_l^{(r)} = 2^{h-r} \ (\forall\, l, r \mid b_l^{(r)} \in B_l) \tag{3.7}$$

In this function, h is the index of the lowest ranked bid among all the bids ($h = \max_l |B_l|$), and therefore the minimum bid weight value, w_{min} is 1. This function ensures that the weights of the bids that have the same rank among the students are equal and for each bid $b_l^{(r)} \in B_l$, the weight of the bid is greater than the sum of the weights of the lower ranked bids in the same set. Therefore, this bid weight function enables the model to satisfy as many higher ranked bids as possible while preserving fairness among the students.

Defining the weight values w_{ij} for the requested courses is straightforward. As for the bids, each student declares the courses in the request set such that the more preferred course comes before the less preferred course (i.e. $\forall i \,|\, b_i \in B : a_{i1} \succ a_{i2} \succ \ldots \succ a_{ip}$). Since for each bid at most one requested course can be added to the students schedule in the optimum solution, the only requirement for weights of the requested courses is to ensure that the more preferred requested course has higher weight value than that of the less preferred course. Therefore, the weight value of the j^{th} requested course is simply defined as:

$$w_{ij} = m - j + 1 \; (\forall \, i \mid b_i \in B) \tag{3.8}$$

where m is the number of courses ($m = |C|$). This simple function ensures that the requested courses with the same rank have equal positive weights among all the bids.

As seen from the objective function given in Eq.(3.1), there are two objectives of the model: the first objective is to maximize the number of satisfied bids according to bid weight values and the second objective is for each satisfied bid to add one requested course with the maximum possible weight. In order to maximize the total satisfaction of students, the first objective is favored against the second objective so that when finding the optimum solution among the feasible solutions, the solution with the maximum sum of the bid weights is chosen as the optimum. However, if there are multiple solutions with the same maximum sum of the bid weights, then the solution with the maximum sum of the weights of the requested courses is chosen among these solutions. In order to provide this feature, the ranges of these two types of weights should be separated in order to cancel the effects of the latter to the former. For this purpose, a constant factor α for scaling the sum of the weights of the requested courses is introduced in the objective function. The α factor is defined as follows:

$$\alpha = \frac{1}{(|B| - 1) \cdot m} \tag{3.9}$$

The following proposition proves that using these weight functions and the α value, the first objective of the model is favored against the second objective.

Proposition 3.1. *Given any two different nonzero feasible solutions with different sums of the bid weights for the direct barter model, the solution with higher sum of the bid weights has higher objective value according to Eq.(3.1) independent of weights of the requested courses.*

Proof. Let B_1 and B_2 be two sets of satisfiable bids that correspond to any two nonzero feasible solutions for the direct barter model, and z_1 and z_2 be the corresponding objective values such that

$$z_1 = \sum_{\forall i \,|\, b_i \in B_1} \left(w_i + \alpha \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij} = 1} w_{ij} \right) \tag{3.10}$$

$$z_2 = \sum_{\forall i \,|\, b_i \in B_2} \left(w_i + \alpha \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij} = 1} w_{ij} \right) \tag{3.11}$$

We will show that if the sum of the bid weights of B_1 is greater than the sum of the bid weights of the B_2, then the objective value z_1 is always greater than the objective value z_2. Therefore, we will be proving that for any two feasible solutions, the solution with the higher sum of the bid weights has higher objective value independent of the sum of the weights of the requested courses.

Suppose that the sum of the bid weights of the B_1 is greater than the sum of the bid weights of the B_2,

$$\sum_{\forall i \,|\, b_i \in B_1} w_i > \sum_{\forall i \,|\, b_i \in B_2} w_i \tag{3.12}$$

Then, the difference between the sums of the bid weights of B_1 and B_2 is a positive integer:

$$\left(\sum_{\forall i \,|\, b_i \in B_1} w_i - \sum_{\forall i \,|\, b_i \in B_2} w_i \right) = k \quad (k \in \mathbb{Z}^+) \tag{3.13}$$

The difference between the objective values z_1 and z_2 is

$$z_1 - z_2 = k + \alpha \cdot \left(\sum_{\forall i \,|\, b_i \in B_1} \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij}=1} w_{ij} - \sum_{\forall i \,|\, b_i \in B_2} \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij}=1} w_{ij} \right) \tag{3.14}$$

Since exactly one requested course is added for each satisfied bid, the lower bound for the sum of the requested course weights of B_1 is

$$\sum_{\forall i \,|\, b_i \in B_1} \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij}=1} w_{ij} \geq \min_{i,j} w_{ij} = 1 \tag{3.15}$$

Because of the conditional proof assumption in Eq.(3.12), B_1 cannot be a subset of B_2, and therefore the upper bound for the sum of the requested course weights of B_2 is

$$\sum_{\forall i \,|\, b_i \in B_2} \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij}=1} w_{ij} \leq (|B| - 1) \cdot \max_{i,j} w_{ij} = (|B| - 1) \cdot m \tag{3.16}$$

Therefore,

$$\left(\sum_{\forall i \,|\, b_i \in B_1} \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij}=1} w_{ij} - \sum_{\forall i \,|\, b_i \in B_2} \sum_{\forall j \,|\, (a_{ij}, w_{ij}) \in R_i \wedge y_{ij}=1} w_{ij}\right) \geq 1 - (|B| - 1) \cdot m \tag{3.17}$$

Using $\alpha = 1/(|B| - 1) \cdot m$, the smallest difference between the objective values is:

$$z_1 - z_2 \;\geq\; 1 + \frac{1}{(|B| - 1) \cdot m} \cdot [1 - (|B| - 1) \cdot m] \tag{3.18}$$

$$z_1 - z_2 \;\geq\; \frac{1}{(|B| - 1) \cdot m} \tag{3.19}$$

$$z_1 - z_2 \;\geq\; 0 \tag{3.20}$$

Therefore, the solution with the higher sum of the bid weights has higher objective value independent of the sum of the weights of the requested courses. □

In general, as the number of courses with remaining quotas increases, the number of solutions with identical values in the first summand of the objective function is likely to increase. The reason is that for satisfiable bids there will be more than one alternative requested course that can be added. Hence, the weight mechanism for the requested courses and α factor mechanism will play an important role for increasing the satisfaction of students in these cases by enabling their favored courses to be added to their schedule.

3.4. Solution Procedure

Since the direct barter model can be formulated using integer programming, its problem instances can be solved using general purpose integer programming solvers. However, resemblance of this model to the used car salesman problem (UCSP) in [105] and the polynomial time barter models in [81, 82] motivated us to search for a network flow based solution. Because of the bid weights and the recursive definition of bid satisfiability that causes circular patterns in the solution like UCSP, we modeled the direct barter problem as a minimum cost flow problem [106]. The minimum cost flow problem is defined as follows: let $N(V, A, l, u, c, b)$ denote a network with node set V, arc set A, lower bound $l(v, w)$, capacity $u(v, w)$, cost $c(v, w)$ values for each arc $(v, w) \in A$, and supply/demand values $b(v)$ for each node $v \in V$. Let $x(v, w)$ represent the flow on arc $(v, w) \in A$. The minimum cost flow problem is defined as follows:

$$\text{Minimize} \sum_{\forall v,w \,|\, (v,w)\in A} c(v, w) \cdot x(v, w) \tag{3.21}$$

$$\text{such that} \sum_{\forall w \,|\, (v,w)\in A} x(v, w) \; - \sum_{\forall w \,|\, (w,v)\in A} x(w, v) \; = \; b(v) \;\; (\forall v \,|\, v \in V) \tag{3.22}$$

$$l(v, w) \leq x(v, w) \leq u(v, w) \;\; (\forall v, w \,|\, (v, w) \in A) \tag{3.23}$$

where $\sum_{\forall v \,|\, v\in V} b(v) = 0$.

To help us in defining the network in our problem formally, we first introduce a set P, called *restriction-pairs set*, which consists of course-student pairs. The set P is defined as follows: $P = \{(c_k, s_l) \,|\, c_k \in C \wedge s_l \in S \wedge (\exists i, i', j, j' \,|\, i \neq i' \wedge b_i, b_{i'} \in B_l \wedge (c_k, w_{ij}) \in R_i \wedge (c_k, w_{i'j'}) \in R_{i'})\}$. Thus, each pair (c_k, s_l) in P indicates that the student s_l requests the course c_k in his at least two different bids, and therefore the student s_l must be prevented from adding the course c_k more than once in the final solution. Based on this definition, the minimum cost flow network can be constructed as follows:

- The set of nodes, V, consists of four types of nodes:
  (*i*) a course node c_k for each course $c_k \in C$,
  (*ii*) a special node $CENTER$ that represents the null course to be dropped for add bids,
  (*iii*) a bid node b_i for each bid $b_i \in B$,
  (*iv*) a restriction node r_{kl} for each $(c_k, s_l) \in P$ for preventing the student s_l from adding the course c_k more than once.

- The set of arcs, A, consists of seven types of arcs:
  - (i) an arc $(c_k, CENTER)$ for each course $c_k \in C$ with capacity equal to q_{c_k} and cost equal to 0 which represents the remaining quota of the course c_k,
  - (ii) an arc $(CENTER, c_k)$ for each course $c_k \in C$ with capacity equal to $+\infty$ and cost equal to ϵ,
  - (iii) an arc (d_i, b_i) for each *barter* and *drop-unless-barter* bid $b_i = (d_i, w_i, R_i)$ with capacity equal to 1 and cost equal to $-w_i$,
  - (iv) an arc $(CENTER, b_i)$ for each *add* bid $b_i = (c_\emptyset, w_i, R_i)$ with capacity equal to 1 and cost equal to $-w_i$,
  - (v) for each course $c_k \in C$ and for each student $s_l \in S$ such that $(c_k, s_l) \in P$:
    - (a) an arc (b_i, r_{kl}) for each bid $b_i = (d_i, w_i, R_i) \in B_l$ if there exists a tuple $(c_k, w_{ij}) \in R_i$ with capacity equal to 1 and cost equal to $-\alpha \cdot w_{ij}$, (i.e. $\forall i, j, k, l \mid s_l \in S \wedge c_k \in C \wedge b_i \in B_l \wedge (a_{ij}, w_{ij}) \in R_i \wedge c_k = a_{ij} \wedge (c_k, s_l) \in P :$ an arc (b_i, r_{kl})),
    - (b) an arc (r_{kl}, c_k) with capacity equal to 1 and cost equal to 0, (i.e. $\forall k, l \mid s_l \in S \wedge c_k \in C \wedge (c_k, s_l) \in P :$ an arc (r_{kl}, c_k)),
  - (vi) for each course $c_k \in C$ and for each student $s_l \in S$ such that $(c_k, s_l) \notin P$:
    - (a) an arc (b_i, c_k) for each bid $b_i = (d_i, w_i, R_i) \in B_l$ if there exists a tuple $(c_k, w_{ij}) \in R_i$ with capacity equal to 1 and cost equal to $-\alpha \cdot w_{ij}$, (i.e. $\forall i, j, k, l \mid s_l \in S \wedge c_k \in C \wedge b_i \in B_l \wedge (a_{ij}, w_{ij}) \in R_i \wedge c_k = a_{ij} \wedge (c_k, s_l) \notin P :$ an arc (b_i, c_k))
  - (vii) an arc $(b_i, CENTER)$ for each drop-unless-barter bid $b_i \in D$ with capacity equal to 1 and cost equal to w_i.

Lower bounds $l(v, w)$ for all arcs $(v, w) \in A$ are set to 0. Similarly, there is no supply or demand for any node in the network, and therefore $b(v) = 0$ for every node $v \in V$. Note that ϵ which is used as the cost of the arcs of type (ii) is the smallest possible positive number representable on the computer ($\epsilon > 0$). It is used to prevent zero cost cycles.

The minimum cost flow network for the example problem given in Section 3.2 and its solution can be seen in Figure 3.1. As stated earlier, all drop bids should always be

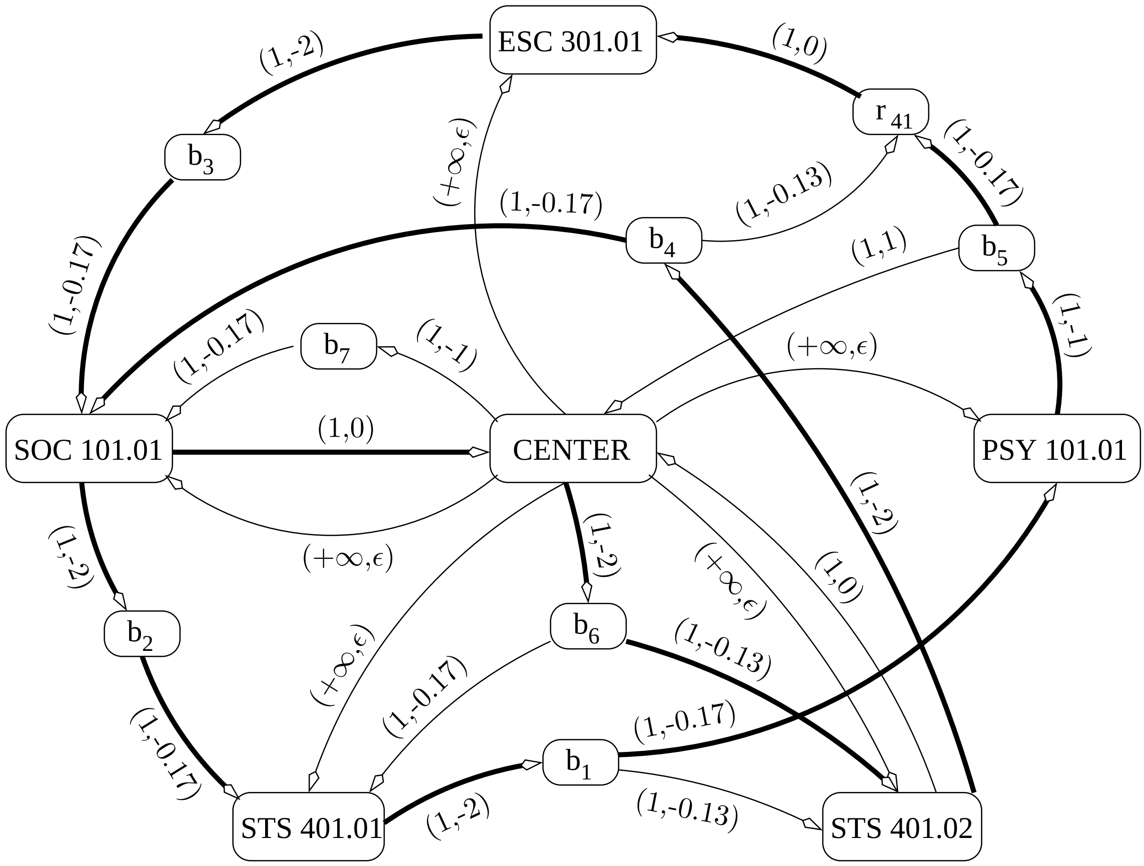


Figure 3.1. Minimum cost flow network of the example given in Section 3.2 with (capacity,cost) values on the arcs. The solution is shown with the bold arcs where one unit of flow passes in each bold arc.

satisfied and since they are always part of the solution, they need not to be included in the network. Therefore, before constructing the network, as a preprocessing step all drop bids are marked as satisfied and the remaining quotas of the courses are increased accordingly. For instance, if there are z drop bids for the course c_k, after satisfying these bids the remaining quota of the course becomes $q_{c_k} + z$. So, when presenting the network, we assume that there will be no drop bids in the bid set B and the set of remaining quotas Q is adjusted accordingly. This simple preprocessing step eliminates drop bids and reduces the network size. Therefore, for the example problem, the drop bid 8 is marked as satisfied beforehand and the quota of the course SOC 101.01 is increased by one.

Verifying the correctness of the described network is straightforward. The arcs of type (*iii*) and (*iv*) represent the binary decision variables x_i for barter and add bids respectively. Therefore, additive inverses of the bid weights, $-w_i$, are used as the costs of these arcs. This statement is also valid for the drop-unless-barter bids on condition that there is no flow on the corresponding arc of type (*vii*). However, if there is a flow passing through both the arc of type (*iii*) and the arc of type (*vii*) for a drop-unless-barter bid, this means that only the drop part of the bid is satisfied. In this case, this bid is considered as unsatisfied in accordance with the integer programming formulation in Section 3.3 and the sum of the costs of the corresponding arcs of type (*iii*) and (*vii*) are zero. As in the case of the bid weights, additive inverses of the weights of the requested courses, $-\alpha \cdot w_{ij}$, are used as the costs of type ($v.a$) and ($vi.a$) arcs that represent binary decision variables y_{ij}. Costs of the other arcs are zero. Therefore, minimization of the cost yields maximization of the sum of the weights. In order to satisfy a bid, one unit of flow must flow from the node representing the course to be dropped (for add bids, from the center node) to the node representing the course to be added. The capacity limits on type (*iii*) and (*iv*) arcs ensure that only one of the requested courses is added for each satisfied bid. Also, the capacity limits on type ($v.b$) arcs prevent students from registering for a course more than once. The arcs of type (*i*) represent the remaining quotas of the respective courses and type (*ii*) arcs allow satisfaction of barter bids when the courses to be dropped are not requested by any other satisfied bid. Finally, the arcs of type (*vii*) allow barter bids which are marked as drop-unless-barter to drop the course if the bartering is not possible. Quota restrictions of the courses are enforced using the flow conservation property of the network nodes c_k that correspond to the courses. Outgoing flow from a course node is restricted with the remaining quota (adjusted value according to the drop bids in the preprocessing step) plus the number of satisfied bids that drop the course.

When the minimum cost flow is found on the network, winning bids can be determined by checking flow on arc types (iii) and (iv) for barter and add bids respectively. If the flow on an arc of these types is 1, it shows that corresponding barter or add bid is satisfied and it is in the optimum solution. Similarly, the arc of type ($v.a$) or ($vi.a$) that originates from the winning bid determines the course to be added for that bid.

Since there can only be one arc with one unit of flow among these arcs, the head of this arc shows the course to be added for the winning bid.

There are strongly polynomial algorithms for solving minimum cost network flow problems such as the minimum mean cycle-canceling algorithm with time complexity $O(|V|^2|A|^3 log|V|)$ [107] and the enhanced capacity scaling algorithm with time complexity $O((|A|log|V|)(|A|+|V|log|V|))$ [106]. Since the minimum cost flow network for a direct barter problem instance can be constructed in polynomial time using the above algorithm, the optimum solution of the direct barter problem given in Section 3.3 can also be found in polynomial time.

3.4.1. Extending the Functionality of the Restriction Nodes

The function of the restriction nodes in the network is to prevent a student from registering the same course more than once. However, the function of the these nodes can be extended so that instead of a single course, any number of disjoint course restriction sets can be defined for each student such that the student can register for at most one of the courses in this set. This is especially useful when a student requests more than one section of the same course, e.g. CMPE 230.01 and CMPE 230.02, or a set of conflicting courses in his at least two bids. For instance, assume that a student submits the following bids:

$$\texttt{CMPE 250.01} \xrightarrow{w_1} \{(\texttt{CMPE 220.01}, w_{11}), (\texttt{CMPE 230.01}, w_{12}), (\texttt{CMPE 230.02}, w_{13})\}$$
$$\texttt{CMPE 240.01} \xrightarrow{w_2} \{(\texttt{CMPE 230.01}, w_{21}), (\texttt{CMPE 230.02}, w_{22}), (\texttt{CMPE 322.01}, w_{23})\}$$

It is clear that the student cannot register for two different sections of CMPE 230 at the same time. Therefore, in order enforce this restriction, we define a course restriction set that consists of restricted courses CMPE 230.01 and 230.02. Instead of using separate restriction nodes for these restricted courses, a split restriction node pair is introduced for this set as shown in Figure 3.2. Then, for each bid of the student that requests at least one of the courses in this set, an arc is drawn from the corresponding bid node to the first node of the pair. The capacity of this arc is set to one and the cost of

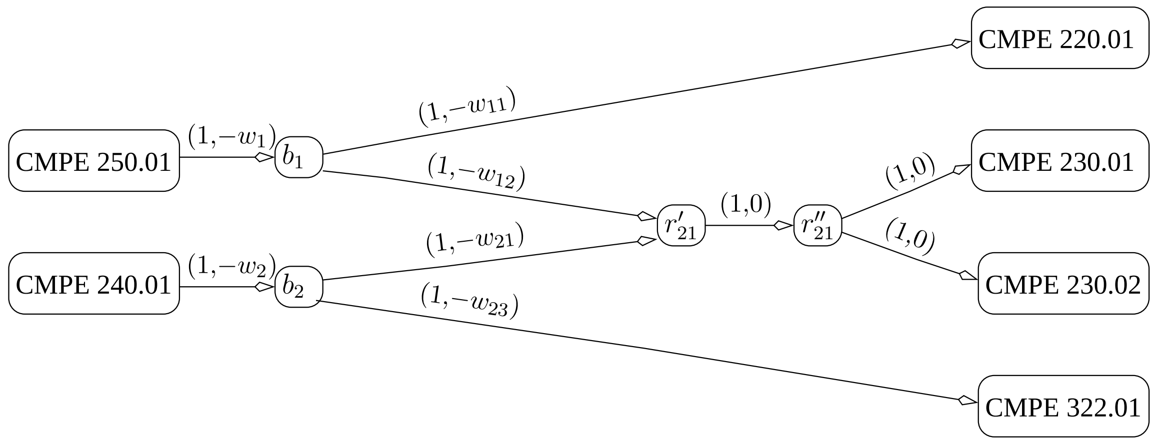


Figure 3.2. Network for illustrating the usage of the restriction nodes for a course restriction set with (capacity,cost) values on the arcs.

this arc is set to the additive inverse of the weight of the highest ranked restricted course in the bid. Additionally, an arc is drawn between the pair nodes with one unit of capacity and zero cost. This arc limits the number of restricted courses to be added to one. Finally, for each restricted course, an arc is drawn from the second node of the pair to the corresponding course node with a capacity of one and cost of zero. This procedure is repeated by introducing a restriction node pair for each course restriction set defined.

3.5. Experimental Results

In order to estimate the real-world performance and the quality of the solutions of our barter model, we developed a test case generator based on real-world student registration data obtained from the Boğaziçi University registration system [102] for the academic year 2008-2009. Since the number of undergraduate students in our university is relatively small, that is 7,095, we generated a statistical profile using the actual registration data and determined the parameters of the test case generator accordingly so that it is capable of generating test cases for arbitrary number of students. The parameters of the test case generator and its source code can be found in [108].

Table 3.2. Running times of the network solver for the test cases (*seconds*). (*) This row presents the results for Boğaziçi University data.

		Avg. #	*Running Time (s)*	
# Students	*# Courses*	*Bids*	*mean*	*stdev*
*7,095	1,158	22,700	0.57	0.03
10,000	1,632	32,090	1.07	0.04
25,000	4,080	80,012	3.85	0.13
50,000	8,160	160,099	9.00	0.29
100,000	16,321	320,675	20.44	0.75

We conducted two different experiments on a dedicated 64 bit Intel Xeon 2.66 GHz workstation with eight GB memory using Linux operating system. We used CS2 software which contains a solver for the minimum cost flow problems based on scaling push-relabel algorithm [109, 110].

In the first experiment, a group of 20 test cases are generated for each selected number of students ranging between 7,095 and 100,000. The average running time of the network solver for each group and the corresponding standard deviation are presented in Table 3.2 and the associated plot is depicted in Figure 3.3. As seen from the results, the solver finds the solution of the problem instances with 100,000 students and approximately 320,000 bids in less than 21 seconds which is quite small. For the case of our university, on the other hand, each instance is solved in less than one second.

In the second experiment, the solutions of our barter model are compared with the currently used FCFS based system. The purpose of this experiment is to present the improvement in the optimum solutions of the test cases over the FCFS approach under different occupancy rates for the courses. Thus, the importance of the introduced bartering mechanism and the weighting mechanism could be observed. In order to simulate the FCFS system, a random permutation of the bids in each test case is

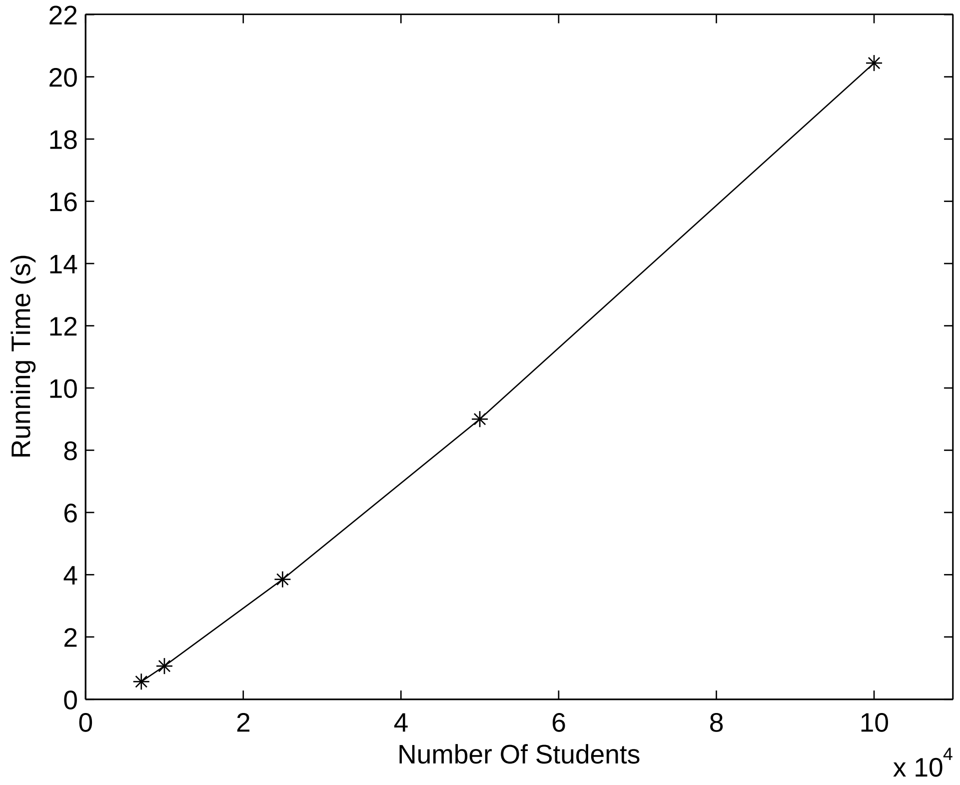


Figure 3.3. Graph of the number of students vs. running time (seconds) of the network solver.

generated by preserving the preferred order of bids of each student. The bids in the permuted list are processed one by one, simulating the way the students submit the bids to the registration system. The processing step is straightforward; for each bid, the remaining quotas of the courses in the request set is checked in the order of students' preferences and if one empty slot is found, the course is added to the schedule of the student. If the processed bid is a barter bid, then the course to be dropped is also removed from his schedule and the remaining quota of that course is increased by one. The whole simulation process is repeated up to five times for the unsatisfied bids.

The second experiment is conducted for two different numbers of students, that is 7,095 and 50,000, where the number of courses are 1,158 and 8,160 respectively, For each number of students, the ratio of the courses without remaining quota to the number of all courses, called p, is varied between 0.20 and 0.99 meaning that approximately

Table 3.3. Improvements in the solutions of the barter model over the FCFS model. (*) This row presents the results for Boğaziçi Univ. data.

			# Satisfied Bids (mean / stdev)			*# Satisfied Students (mean / stdev)*		
# Stu.	*(p)*	*# Bids*	*FCFS*	*Barter*	*Impr.*	*FCFS*	*Barter*	*Impr.*
7,095	0.20	22,757	18,367 / 148	20,642 / 129	**12.4%**	6,719 / 17	6,828 / 15	**1.6%**
	*0.28	22,725	17,823 / 176	20,409 / 153	**14.5%**	6,674 / 17	6,822 / 16	**2.2%**
	0.40	22,685	16,847 / 184	19,905 / 155	**18.2%**	6,605 / 23	6,820 / 12	**3.2%**
	0.60	22,761	14,224 / 260	18,656 / 140	**31.2%**	6,322 / 39	6,814 / 14	**7.8%**
	0.80	22,729	9,174 / 437	17,212 / 190	**88.0%**	5,329 / 128	6,812 / 16	**27.9%**
	0.90	22,723	5,103 / 497	16,300 / 116	**222.5%**	3,737 / 267	6,760 / 23	**81.8%**
	0.95	22,745	2,660 / 503	16,050 / 163	**522.9%**	2,249 / 356	6,671 / 24	**203.4%**
	0.99	22,726	556 / 123	15,569 / 151	**2,827.9%**	537 / 114	6,580 / 25	**1,178.1%**
50,000	0.20	160,226	129,495 / 513	145,312 / 405	**12.2%**	47,285 / 53	48,040 / 47	**1.6%**
	0.40	160,312	118,538 / 595	140,396 / 407	**18.4%**	46,469 / 68	48,033 / 37	**3.4%**
	0.60	160,319	100,492 / 1,138	131,704 / 506	**31.1%**	44,625 / 167	48,039 / 41	**7.7%**
	0.80	160,274	64,312 / 1,330	121,217 / 416	**88.6%**	37,416 / 383	48,017 / 55	**28.3%**
	0.90	160,052	35,624 / 1,321	115,162 / 326	**223.7%**	26,203 / 691	47,675 / 57	**82.1%**
	0.95	160,477	18,984 / 994	113,228 / 345	**497.9%**	16,084 / 731	47,010 / 72	**192.8%**
	0.99	160,276	3,597 / 507	109,539 / 298	**3,003.1%**	3,480 / 476	46,300 / 43	**1,254.2%**

$(100{\cdot}p)\%$ of the courses have no remaining quota. For each configuration, again a group of 20 test cases are generated. The results of this experiment are given in Table 3.3.

For the test cases with 7,095 students, assuming that 20% of the courses are full, the number of satisfied bids in the solution found by the barter model is approximately 12% higher than that of the FCFS based system. For the case of our university where approximately 28% of the courses are full, the barter model provides 15% better results. It is remarkable that the improvement percentage for the number of satisfied bids

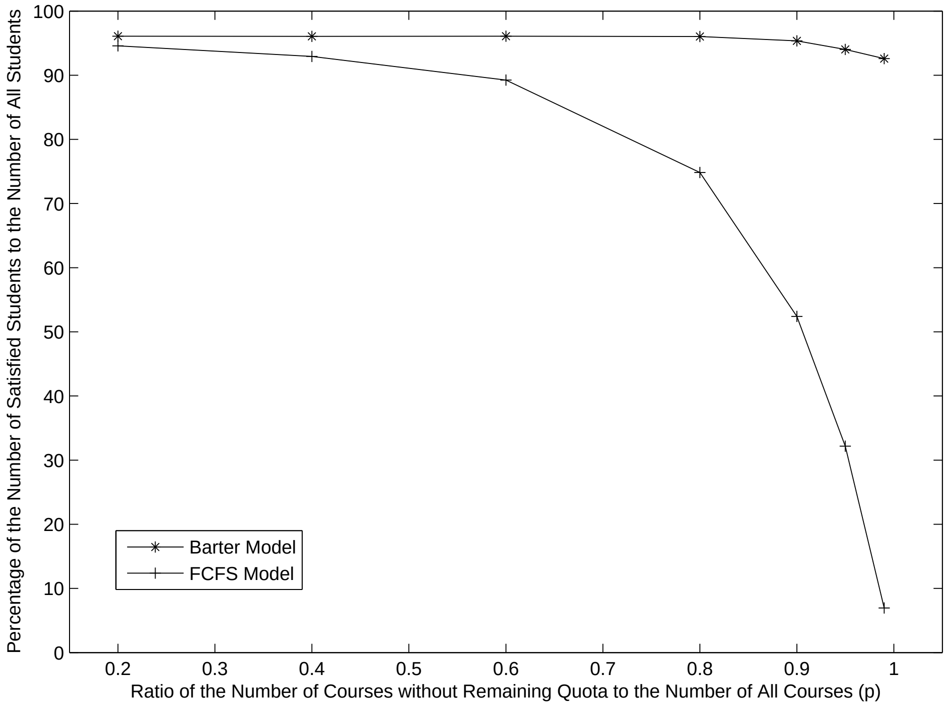


Figure 3.4. Graph comparing the number of satisfied students in the solutions of the barter model and the FCFS model.

increases dramatically as p increases. In the extreme situation where the courses are 99% full, approximately 28 times improvement over the FCFS system is observed. It should also be noted that in the barter model the standard deviations are also very small relative to the mean values so that the quality of the solutions found do not vary much. The results of the test cases with 50,000 students are also very close to the test cases with 7,095 students showing that the model is scalable to the universities with higher number of students without sacrificing the quality of the solutions.

By virtue of the weighting mechanism, the barter model also improves fairness among the students by increasing not only the number of satisfied bids but also the number of satisfied students. As seen from the plot in Figure 3.4, the number of satisfied students in the barter model whose at least one bid is satisfied is approximately 96% of the total number of students and decreases slightly to 93% as p goes to 0.99. However, in the FCFS model, starting from 95%, this ratio drops to 7%.

3.6. Comparison with Previous Work

As stated earlier, the CT and SS problems have been covered extensively in the literature. Some studies have addressed solely the CT problem; several integer linear programming models [111–113] and heuristic methods [111, 112, 114] have been proposed. For the SS problem, Reeves et al. [115] and Willoughby et al. [116] have proposed a mixed integer and network programming models, respectively, and Alvarez-Valdes et al. [117] has applied Tabu search techniques. In [118, 119], on the other hand, unified optimization models that address both problems are presented. There are also open source (for example [104]) and commercial packages (for example [120]) that address these problems. For further details on these problems, the reader is referred to surveys by Burke et al. [98], Carter et al. [99], Lewis [100] and Schaerf [101].

In the remaining parts of this section, we first review Graves et al.'s work [41] in detail which has addressed the student course add/drop process as we do in this chapter. Then, in Section 3.6.2, we discuss the relationship between our model and a somewhat related problem called the *stable college admissions problem* (*SCAP*) [121, 122].

3.6.1. Relationship Between Graves et al.'s Work and the Barter Model

Graves et al. [41] propose an auction based market approach complete with clearing prices for allocating course sections to students. Their model consists of two rounds. In the first registration round, which is called *registration bidding system* (*RBS*), students are granted bidding points (i.e. registration money) which they can use to bid on desired schedules. During this period, students are allowed to place course selections as their bids together with the money they will pay for each schedule. The bids are ranked in descending order of bidding points and are selected if requested course capacities are available. At the end of the registration period, the prices of the courses are determined. Each successful bidder pays the sum of the prices of the assigned courses to him instead of the price he offered for his bid. Therefore, it is possible that a successful bidder may not have enough money in which case a subsidy given by the system covers the deficiency. Subsidies not paid back during add/drop phase as a result of dropping courses are simply forgotten (waived) by the system. Hence, we believe that if this fact is known by the students, then it could easily be abused by offering high prices for their schedules. Since the winning bids can be subsidized and the subsidized amounts can be forgotten, this then introduces fairness problems in Graves et al.'s approach.

In the second round, which is called *drop/add/swap* (*DAS*) round, Graves et al. introduce a course swapping idea. This corresponds to the barter scheme that we propose in this chapter. As in the RBS round, an auction based approach is used. Students submit add, drop and swap bids together with the amounts of bidding points that are carried forward from the RBS round (if any). After the end of the round, a linear program whose objective is to maximize the sum of the bidding points of the satisfied bids is solved. By solving the linear program, the students are assigned to the courses and also the prices of the courses in terms of bidding points (dual prices) are determined. As in the registration round, each student pays the actual amounts of his satisfied bids calculated according to the determined prices of the courses. However, in this case it is stated that subsidies are not allowed. Since there is no formal treatment for this problem except for a simple linear programming example, it is not clear how the restriction for preventing the subsidies is applied in their model. This also prevents one from implementing their model. Although, by the definition of dual prices, the actual amount to be paid for each satisfied bid is bounded by the offered price for that bid, this does not solve the subsidy problem since each student may submit more than one bid. Thus, a further constraint is necessary. In fact, the following proposition shows that introducing a budget constraint in order to prevent subsidies, i.e. the sum of the bidding points of the satisfied bids of a student should be less than or equal to the amount of bidding points owned by that student, to the given linear programming example makes the resulting problem NP-hard. It should be noted that the formal definition of the DAS problem given below is constructed by us according to the linear programming example and the explanations given in [41].

Proposition 3.2. *The decision version of the DAS problem with budget constraint is NP-complete.*

Proof. Let Π be the decision version of the DAS problem with budget constraint. Π is defined as follows: Given

- a set of courses, $C = \{c_1, c_2, \ldots, c_m\}$;
- a sequence of remaining quotas, $Q = \{q_{c_1}, q_{c_2}, \ldots, q_{c_m}\}$ where q_{c_k} is the remaining quota of course c_k ($1 \leq k \leq m$, $q_{c_k} \in \mathbb{N}$);
- a set of students, $S = \{s_1, s_2, \ldots, s_t\}$;
- a set of bids, $B = \bigcup_{l=1}^{t} B_l$ where B_l is the set of bids of a student s_l ($1 \leq l \leq t$) and each bid is denoted by a triplet, $b_i = (d_i, a_i, p_i)$, where d_i is the course to be dropped for *barter* and *drop* bids or the null course, $c_\emptyset$, for *add* bids ($d_i \in C \cup \{c_\emptyset\}$), a_i is the course to be added for *barter* and *add* bids or the null course, $c_\emptyset$, for *drop* bids ($a_i \in C \cup \{c_\emptyset\}$), and p_i is the amount of bidding points offered by the student for bid b_i ($1 \leq i \leq n = |B|$, $p_i \in \mathbb{N}$);
- a set of bid restrictions, $L = \{l_1, l_2, \ldots, l_z\}$ that consists of mutually disjoint subsets of bids, $l_y \subseteq B$ ($1 \leq y \leq z$), such that at most one of the bids in l_y can be satisfied (e.g. a student may put a restriction on two of his add bids so that only one of them can be satisfied or the system may enforce a restriction on two barter bids of a student in which the same course is dropped);
- a sequence of bidding points owned by students, $F = \{f_1, f_2, \ldots, f_t\}$ where f_l is the amount of bidding points owned by student s_l ($1 \leq l \leq t, f_l \in \mathbb{N}$);
- a positive integer K;

is there a subset $B' \subseteq B$ such that the following inequalities are satisfied?

$$\sum_{\forall i \,|\, b_i \in B'} p_i \geq K \tag{3.24}$$

$$\sum_{\forall i \,|\, b_i \in (B_l \cap B')} p_i \leq f_l \qquad (\forall l \,|\, s_l \in S) \tag{3.25}$$

$$\sum_{\forall i \,|\, b_i \in B' \wedge a_i = c_k} 1 - \sum_{\forall i \,|\, b_i \in B' \wedge d_i = c_k} 1 \;\leq\; q_{c_k} \qquad (\forall k \,|\, c_k \in C) \tag{3.26}$$

$$\sum_{\forall i \,|\, b_i \in (l_y \cap B')} 1 \leq \; 1 \qquad (\forall y \,|\, l_y \in L) \tag{3.27}$$

In this formulation, Eq.(3.24) ensures that the sum of the offered bidding points for all satisfied bids (B') is greater than or equal to the positive integer K. Eq.(3.25) is the budget constraint which prevents the sum of the bidding points of the satisfied bids of a student from exceeding the amount of bidding points owned by that student. The quota restrictions are enforced in Eq.(3.26). For each course, the number of students who drop the course plus the remaining quota of the course should be greater than or equal to the number of students who add the course. Finally, Eq.(3.27) ensures that bid restrictions are applied such that for all y $(1 \leq y \leq z)$, at most one of the bids in $l_y \in L$ is satisfied.

If we have a certificate that consists of $B' \subseteq B$, this certificate can be verified in polynomial time by checking Eq. (3.24-3.27). Therefore Π is in NP.

Next, we present a polynomial time transformation from the subset sum problem. Let Π' be the subset sum problem (see for example: [123, p. 73] and [50, p. 247]) which is defined as follows: Given a finite set U, a weight value $w(u_i) \in \mathbb{Z}^+$ for each $u_i \in U$ $(1 \leq i \leq |U|)$, and positive integers C and K, is there a subset $U' \subseteq U$ such that

$$\sum_{\forall i \,|\, u_i \in U'} w(u_i) \geq K \tag{3.28}$$

$$\sum_{\forall i \,|\, u_i \in U'} w(u_i) \leq C \tag{3.29}$$

Let $\Pi'(U, w(u_i), C, K)$ be an instance of the subset sum problem. It can be transformed to Π in polynomial time as follows: let the set of courses C consist of $|U|$ courses $(C = \{c_1, c_2, \ldots, c_{|U|}\})$ and the remaining quotas of all the courses be one $(q_{c_k} = 1,\ k = 1, 2, \ldots, |U|)$. The students set S consists of one student $(S = \{s_1\})$ and for each $u_i \in U$, the student s_1 submits an add bid b_i requesting the course c_i with a price of $w(u_i)$ $(i = 1, 2, \ldots, |U|)$. The amount of bidding points of the student s_1 is C $(F = \{C\})$. Let the set of bid restrictions, L, be empty. Since in each bid exactly one unique course is requested and the remaining quotas of all the courses are 1, Eq. (3.26) always holds independent of the set B'. Thus, for the transformed problem instances, Eq. (3.24-3.27) reduce to the inequalities of the subset sum problem:

$$\sum_{\forall i \,|\, b_i \in B'} w(u_i) \geq K \tag{3.30}$$

$$\sum_{\forall i \,|\, b_i \in B'} w(u_i) \leq C \tag{3.31}$$

and therefore, the solution of a problem instance of Π is also the solution of the corresponding problem instance of Π', and the solution of a problem instance of Π' is also the solution of the problem instance of Π.

Since Π is in NP and the subset sum problem is NP-complete, the decision version of the DAS problem with dynamic credit constraint is NP-complete. □

Besides the subsidy and the associated unfairness problems in Graves et al. approach, students are also responsible for deciding the prices of the bids according to certain upper and lower bounds. However, determining the prices can be cumbersome for the students because of the combinatorial nature of the model. Furthermore, since the remaining points of the students are transferred to the next semester, decision making will become tougher since the students should also consider the following semesters. Finally, we note that although the students' perception of the quality of their schedules is not quantified, Graves et al. estimate one percent increase in the quality of schedules using their model.

3.6.2. Relationship with the Stable University Admissions Problem

The preference based ordering of bids and requested courses indicates a relationship between the direct barter model and the stable college admissions problem (SCAP) [121, 122]. In the SCAP, there are two sets of agents, colleges and students. Each college has strict preferences over the students and can accept a limited number of students. Each student, on the other hand, can enroll to only one college and has also strict preferences over the colleges. The SCAP is defined as finding a matching of students to colleges, called a *stable allocation*, such that no unmatched pair of opposite agents would simultaneously be better off if they were matched together.

In general, the SCAP cannot be used to solve our barter model. However, if we consider a simple special case of our model in which *only add bids are allowed* and *the weights of the bids are unique*, then this simple problem can be reduced to the SCAP as follows (note that the latter requirement cannot be satisfied when the bid weight function in Eq. (3.7) is used. Therefore, even the instances of direct barter model consisting of add bids cannot be reduced to the SCAP.): let the set of bids B and the set of courses C in our barter model map to the row agents I (students), and the column agents J (colleges) in [122] respectively. The function $\pi(i, j)$ in the SCAP indicates whether student i wants to admit to college j or not. Therefore, for each bid $b_i \in B$ and for each course $c_i \in C$ in our model, if c_i is requested in bid b_i, then $\pi(i, j)$ is set to 1. Otherwise, it is set to 0. For each bid $b_i \in B$ in our model, the weights of the requested courses correspond to the strict college preference of the student i. For each course $c_k \in C$ in our model, the weights of the bids that request the course c_k correspond to the strict student preference of the college j. In the SCAP, each student i can enroll at most $s(i)$ colleges and each college j can accept $d(j)$ students. In our model, since only one course can be assigned to a bid b_i, we set $s(i) = 1$. However, a course c_k can be assigned to q_{c_k} students, and therefore we set $d(j) = q_{c_k}$. Then, the "college optimal" deferred acceptance procedure [121] produces a stable allocation in which the courses are assigned to the bids in such a way that the bid weights are favored against the weights of the requested courses. However, the stable allocation found by the deferred acceptance procedure can be different from the allocation found by the direct barter model since the direct barter model does not seek a stable allocation but an allocation that maximizes total satisfaction of students. For instance, assume that the following two add bids are submitted:

Student 1 (bid 1): $c_\emptyset \rightarrow \{\texttt{course } A, \texttt{ course } B\}$

Student 2 (bid 2): $c_\emptyset \rightarrow \{\texttt{course } A\}$

Suppose that the weight of the first bid is higher than that of the second bid. Then, the deferred acceptance procedure would assign course A to Student 1 which is the only stable allocation. However, using the direct barter model both bids would be satisfied and Student 1 would get course B and Student 2 would get course A.

3.7. Conclusion

In this chapter, we have modeled the course add/drop process as a direct bartering problem in which add/drop requests appear as bids. We formulated the resulting problem as an integer linear program, and then showed that our problem can be solved in polynomial time as a minimum cost network flow problem. In our model, we also introduced a two level weighting mechanism that enables students to express their preferences for their bids and for the requested courses. The weighting mechanism also improves fairness among the students. As demonstrated with the experimental results, the minimum cost flow network solvers can solve problem instances for a typical university within seconds. Hence, our algorithms can be deployed in universities with hundreds of thousands of students without worrying about execution times. The experimental results also show significant improvement in the quality of the solutions over the currently used FCFS based system while preserving the fairness, and hence we believe that the direct bartering model will greatly improve the satisfaction of students in the universities.

As a final note, the real-life performance of our model also motivates us to apply the direct barter approach to different application areas. For instance, an electronic exchange that facilitates bartering e-media such as e-books, music and movies on the Internet can be designed in a similar manner.

4. MULTI-UNIT DIFFERENTIAL AUCTION BARTER MODEL

4.1. Introduction

E-markets offer services such as directory listings and searching of goods or services and transaction support. Secure, fast and reliable communication media in e-markets help their customers in forming new trading partnerships and reducing communication costs. However, for increasing the trading volume in the e-markets, besides these services, a matching/brokering service is necessary between sellers and buyers in order to satisfy their possibly complex supply and demand requirements [124]. Auction based approaches have been proposed for this task [125]. For instance, Özturan [105] proposes a hybrid differential auction-barter (DAB) model which extends the single-unit DA institution for increasing the trading volume of online used vehicle auctions such as autobytel.com, the eBay Motors and the Yahoo! Autos. In these auctions, buyers place bids for the vehicles they want to purchase and sellers place asks for the cars they want to sell. Although bidders can submit both sale and purchase bids, these bids are considered as independent and they are not allowed to put conditional restrictions for their bids. This discourages bidders who are willing to sell their current vehicles only if they are able to purchase other vehicles. For instance, assume that a bidder currently owns a car of brand A and wants to exchange (barter) his car for a car of brand B while paying at most $5,000. He must first determine reservation prices for both car A and car B. The *reservation price* is the minimum (maximum) price at which the bidder is willing to sell (buy) an item. Assuming that the reservation price of the bidder for his car A is $10,000, he will place an ask of $10,000 for his car A and a bid of $\$10,000 + \$5,000 = \$15,000$ for car B. Depending on the auction outcome, he may lose his car without getting a replacement car or be forced to pay $15,000 for an additional vehicle without selling his current vehicle, resulting in a $10,000 budget deficit. In order to prevent this inefficiency, the DAB model extends the single-unit DA so that the bidders are also allowed to place *barter bids* in which an item can be bartered for another item. Besides, if the values of the items are not considered to be same, then the bidders are also allowed to declare an amount of money, called *differential price*, to be given or taken along with their barter request. For instance, for the above case, the bidder can submit a bid declaring that he wants to exchange his car A for car B and pay \$5,000. This bid is represented using the following notation:

$$\{\text{Car A} + \$5{,}000\} \rightarrow \{\text{Car B}\}$$

Since the DAB model is designed especially for online used car auctions (consumer-to-consumer), it is very unlikely to have more than one unit of an item (e.g. two identical used cars). Therefore, it is designed as a single-unit auction model and does not support multiple identical units of an item. The bidding language is also quite simple. A barter bid consists of exactly two items, an item to be given and an item to be taken, and an associated differential price if any. Similarly, a sale bid or a purchase bid consists of a single item and an associated price the bidder is willing to pay or earn.

In this chapter, we propose a new model called the MUDAB model for e-markets in which multiple instances of commodities are exchanged. The model has been primarily designed for improving the efficiency of a discrete-time DA institution.

The model has three important features which are also explained quantitatively based on an example market scenario in the next section:

(*i*) The model extends both the multi-unit DA and the DAB models so that bidders can put forward barter bids along with the sale and purchase bids for multiple instances of items. This mechanism encourages traders to participate in the e-markets without risking buying of new items unless they sell their items first or vice-versa. Furthermore, in addition to the direct matching of sale and purchase bids as in DA, the model increases the allocative efficiency of the market by extracting barter cycles and sale-barter-purchase chains of any length in the market.

An example scenario for a barter cycle involving three bidders is as follows:

$$\underbrace{\text{Bidder 1}}_{Barter} \xrightarrow{\text{gives item(s) to}} \underbrace{\text{Bidder 2}}_{Barter} \xrightarrow{\text{gives item(s) to}} \underbrace{\text{Bidder 3}}_{Barter} \xrightarrow{\text{gives item(s) to}} \text{Bidder 1}$$

An example scenario for sale-barter-purchase chain involving four bidders is given below:

$$\underbrace{\underset{\text{(earns money)}}{\text{Bidder 1}}}_{Sale} \xrightarrow{\text{gives item(s) to}} \underbrace{\text{Bidder 2}}_{Barter} \xrightarrow{\text{gives item(s) to}} \underbrace{\text{Bidder 3}}_{Barter} \xrightarrow{\text{gives item(s) to}} \underbrace{\underset{\text{(pays money)}}{\text{Bidder 4}}}_{Purchase}$$

Finally, since the MUDAB model is the superset of both the multi-unit DA and the DAB models, the allocative efficiency of the MUDAB model is guaranteed to be at least as good as that of the multi-unit DA and the DAB models.

(*ii*) The model introduces a powerful and flexible bidding language (which is explained in Section 4.4 in detail) which allows bidders to express their complex preferences inside bids. Compared to DAB, the new bidding language allows bidders to combine purchase, sell and exchange requests inside the bids and put restrictions on the number of items to be traded. This feature improves the allocative efficiency of market, and also makes the MUDAB model preferable over the DAB model even for single-unit markets. Furthermore, the new bidding language reduces the total number of bids required for representing preferences of bidders greatly, and hence, reduces the amount of online information exchange between the bidders and the auctioneer.

(*iii*) The optimum allocation in the MUDAB model, that is the optimum set of satisfiable bids, can be determined using the proposed fast polynomial-time algorithm which is based on network flows [106]. Thus, the model can be used in large-scale web-based auctions containing millions of bids without worrying about the running times of the solver.

In the rest of the chapter, we first compare the DA, DAB and MUDAB institutions quantitatively with an example scenario. Section 4.3 presents some electronic commerce applications that our model could be applied. In Section 4.4, a more comprehensive example with which we explain our MUDAB model is given. In Section 4.5, we formulate our model by using linear-integer programming and define the winner determination problem. In Section 4.6, we introduce a minimum cost network flow solution of the winner determination problem. Section 4.7 presents the experimental results and finally, the chapter is concluded in Section 4.8.

4.2. Comparison of DA, DAB and MUDAB Models

In this section, the DA, DAB and MUDAB models will be compared based on an example scenario from a business-to-business paper market involving five bidders. The scenario is presented in Table 4.1. In the paper industry, paper is manufactured in different qualities which we denote as grade A, B or C and is traded in standard sized rolls [69]. In this scenario, Bidder 1 wants to sell 200 rolls of his inventory, and Bidders 4 and 5 want to purchase 100 rolls of grade A and B paper, respectively. Bidders 2 and 3, on the other hand, want to exchange (barter) their 100 rolls of inventory for lower grade paper. The reason could be that lower grade paper could satisfy their requirements and they would like to earn money by exchanging their inventory. The reservation prices of the bidders for different grades of paper are presented in Table 4.2.

If the market was regulated using single-unit or multi-unit DA rules, then the first bidder would want to sell 200 rolls of grade B paper. The reason is that he considers the price of grade B paper higher than the price of grade C paper, and therefore, he would want to make more money. Since there is no bartering option available in DA, the second and the third bidders either would not participate in the market because of the risk of losing their inventory and the risk of having a budget deficit, or they would take these risks and submit sale and purchase bids based on their subjective reservation prices. No decision making would be necessary for the fourth and the fifth bidders, and hence, they would simply submit purchase bids for the paper they want. As shown in Table 4.3, the outcomes of the market would be same for both the single-unit and

Table 4.1. An example scenario for paper auction. The second column indicates the inventories of the bidders before the auction and the third column indicates the requests of the bidders.

	owns	*wants*
Bidder 1	200 rolls of grade B paper and 100 rolls of grade C paper	to sell 200 rolls of his inventory
Bidder 2	100 rolls of grade B paper	to exchange his inventory for 100 rolls of grade C paper
Bidder 3	100 rolls of grade A paper	to exchange his inventory for 100 rolls of grade B paper
Bidder 4	-	to buy 100 rolls of grade A paper
Bidder 5	-	to buy 100 rolls of grade B paper

multi-unit cases. If the second and the third bidders do not participate, then only the sale bid of the first bidder would match the fifth bidder's purchase bid, causing a total of 100 rolls of paper to be traded. However, if they opt to participate, then the third bidder would have to sell 100 rolls to the fourth bidder without getting replacement paper and would have an inventory shortage. The single-unit and the multi-unit cases differ in the number of submitted bids. In first case, the bidders would have to submit one bid for each roll they want to sell and purchase. Therefore, the number of submitted bids in the single-unit case would be 400. However, in the latter case, one bid would be enough for each paper type, and hence, the bidders would have to submit only three bids.

If the market was a DAB institution, then the bidders would be able to place barter bids in addition to the sale and purchase bids. Therefore, Bidders 2 and 3

Table 4.2. Reservation prices of the bidders for the example scenario in Table 4.1. The prices are declared per roll of paper.

	Grade A	Grade B	Grade C
Bidder 1	-	$300	$220
Bidder 2	-	$310	$210
Bidder 3	$400	$290	-
Bidder 4	$430	-	-
Bidder 5	-	$300	-

would submit barter bids in order to exchange their inventory for lower grade paper. The outcome of the market would be same as the DA case since there would be no barter cycles in the market. However, unlike the DA case, the barter bid mechanism would allow Bidders 2 and 3 to participate in the market without taking any risks. The downside of the DAB model is that the number of bids required for representing

Table 4.3. Comparison of the auction models. The outcomes of the single-unit DA, the multi-unit DA, the DAB, and the MUDAB models are presented for the example scenario in Table 4.1.

	Number of Units Traded	*Number of Required Bids*	*Problem Size*	*General Market Clearing Algo. Complexity*
Single-Unit DA	100	400	$n = 7$ (bids)	$O(n \log n)$
Multi-Unit DA	100	3	$n = 7$ (bids)	$O(n \log n)$
DAB	100	40,600	$n =$ 1,302 (nodes) $m =$ 41,200 (arcs)	$O(n^2 m^3 (\log n))$
MUDAB	400	5	$n = 23$ (nodes) $m = 35$ (arcs)	$O(n^2 m^3 (\log n))$

the preferences of the bidders is quite large. Since the model is single-unit, each roll of paper in the market would be introduced as a unique item. Therefore, one sale or purchase bid is required for each roll of paper to be sold or purchased as in the single-unit DA market. For the barter bids, this situation gets worse since the bidders should submit one barter bid for each pair of units of items they want to give and get. For instance, if a bidder wants to exchange m units of an item he owns for m units of another item of which there are n units available in the market, then he must place $m \cdot n$ barter bids. This scenario results in 40,600 bids.

In the MUDAB model, the bidding language allows more complex preferences to be declared. For instance, Bidder 1 can submit a single bid declaring that he wants to sell at most 200 rolls of his inventory without differentiating grade B or C paper. This feature of the model increases the outcome of the market. For this scenario, the MUDAB model finds the optimum allocation of goods which contains the following sale-barter-purchase chain of length four and sale-purchase match:

Bidder 1 (earns \$22,000) [*sale*] —gives 100 rolls grade C paper to→ Bidder 2 (additionally earns \$10,000) [*barter*] —gives 100 rolls grade B paper to→ Bidder 3 (additionally earns \$11,000) [*barter*] —gives 100 rolls grade A paper to→ Bidder 4 (pays \$43,000) [*purchase*]

Bidder 1 (earns \$30,000) [*sale*] —gives 100 rolls grade B paper to→ Bidder 5 (pays \$30,000) [*purchase*]

This allocation clearly satisfies all the requests of the bidders and no bidder pays more or earns less than the amount he declared as his reservation price. In this scenario, the MUDAB model requires only five bids to be submitted since the preference of each bidder can be represented with exactly one bid.

From the computational point of view, the allocation for the single-unit and multi-unit DAs can be found easily (by sorting and matching in log-linear time). In the DAB model, the allocation is found by solving a minimum cost flow problem for the network constructed by the DAB algorithm. Although this problem can also be solved in polynomial-time, the size of the network constructed by the DAB algorithm for multi-unit markets is so large that it may not be possible to find the optimum solution even for small markets. For instance, the network constructed by the DAB model for this small example has 1,302 nodes and 41,200 arcs. However, although the optimum allocation in the MUDAB model is also found by solving the same polynomial-time network flow problem, the size of the networks constructed by the MUDAB algorithm for the multi-unit markets is much smaller than that of the DAB model. The size of the network for this scenario contains only 23 nodes and 35 arcs.

As a final note, we present the taxonomy of barter models in Table 4.4. Although the MUDAB model supports multiple identical units of items, it does not support package (combinatorial) bidding, that is bidding on bundles of items. For instance, a bidder cannot offer to exchange a bundle of items A and B together for a bundle of item C and D. Both single-unit and multi-unit *multi-resource barter problems* which support package bidding are studied by Özturan in [81]. It is proven that both cases of the multi-resource (multi-item) barter problem are NP-Hard. Therefore, the MUDAB model does not incorporate package bidding feature because of the difficulty in finding the optimum allocation computationally. Heuristic approaches also would not be preferred from the economics perspective since in this case, there would be no incentive for the bidders to declare their true utility values inside their bids unless the set of preferences of the bidders is restricted [126]. Being able to calculate the optimum solution in polynomial time opens the way for designing truthful mechanisms [28, 127].

4.3. Electronic Commerce Applications

The MUDAB model is designed particularly as a core component of a matching/brokering service for increasing the quality of service. Although the model is also applicable to small-scale and possibly non-web-based markets, improvements in the trading volume and the allocative efficiency compared to DA can only be observed as the number of barter bids, and thus the barter cycles and chains inside the market increase. For this reason, the primary application area of the model is e-markets in which bidders countrywide or even worldwide can participate and large number of bids can be collected in a short time.

Table 4.4. Taxonomy of barter models. Classification is based on whether identical units of items are allowed in the market and whether the bidding language allows package (combinatorial) bidding.

Allowed Item Exchange Type	Identical Units of Items in the Market: *Single-Unit*	Identical Units of Items in the Market: *Multi-Unit*
Single-Item (one-to-one)	The DAB Model in [105] *(Polynomial)*	The MUDAB Model in This Thesis *(Polynomial)*
Multi-Item (package bidding)	The Single-Unit Multi Resource Bartering Problem in [81] *(NP-hard)*	The Multi-Unit Multi Resource Bartering Problem in [81] *(NP-hard)*

In general, the MUDAB model can be applied to any single-unit or multi-unit e-market in which DA rules are applicable. As an example, the model can be used in a business-to-business e-market where standardized units of overstocked raw materials such as metals, lumber, chemicals, paper, glass, and seed are exchanged. Another non-trivial market example would be vehicle exchanges for car dealerships in which both *new* and used vehicles are traded. For instance, in the National Auto Auction Association's (NAAA) North American member auctions, unlike the consumer car markets, multiple units of both new and used vehicles such as business and government fleets, replaced rental fleets, repossessed vehicles by financial institutions, and off-lease returns and trade-in vehicles are traded. According to the NAAA's 2009 annual review report ([128]), approximately 8.92 million vehicles were sold among the offered 14.6 million vehicles in 2009.

In addition to the commercial multi-unit e-markets, the model can also be used in commercial barter exchanges [129]. A *commercial barter exchange* is an e-platform for businesses to trade their excess business capacities and assets, i.e. goods and services, which are regulated by an intermediary. Examples are Merchants Barter Exchange (merchantsbarter.com), BarterXchange (barterxchange.com) and Turk Barter International A.S. (turkbarter.com). In these exchanges, generally variants of DA institution are utilized. However, instead of using a country's currency, a private currency (e.g. barter dollar) whose value is tied to the corresponding country's currency (e.g. dollar) is used. The MUDAB model can readily be applied to the barter exchanges that use private currency.

The last application domain to be introduced is the e-media exchanges. An *e-media exchange* is an e-market for trading e-media files such as e-books, music, movies and video games between the Internet users. Currently, there are many online stores that sell e-media such as iTunes and Amazon. However, to the best of our knowledge, there is no market for bartering e-media files legally. We propose an e-market in which a central intermediary keeps records of the ownership information (i.e. licenses for e-media files), conducts auctions and regulates transactions among the users. The e-media files purchased by the users could be either served online by the intermediary or downloaded to the users' devices using some kind of digital right protection system. When a user sells a file or exchanges for another file, the intermediary removes the associated license from the user's account or transfers it to the new owner. Since all the transactions including the items are electronic, large-scale auctions with participants from all over the world could be conducted. Therefore, the MUDAB model would be preferred in this kind of exchanges for its fast polynomial-time winner determination algorithm. One final note is that although it may seem like each user will own at most one license for himself, in these exchanges;

- some people may engage in the trading of these files for just making profit, and therefore may own and trade multiple licenses of a file;
- wholesale license suppliers selling multiple units of licenses may also participate.

Even if no user wants to own more than one license for a file, there would be multiple licenses for that file owned by different users in the exchange which would also prevent the original DAB model being used in this kind of e-markets.

4.4. A Comprehensive Example and the Details of the Multi-Unit Differential Auction Barter Model

In this section, we will first introduce the bidding language of the MUDAB model, and then explain the model with an example scenario. In the bidding language of the MUDAB model, a bid is represented as follows:

$$\{\underbrace{(\text{item, exchange limit, price}), \ldots}_{\textit{Ask Set}}\} \xrightarrow{\text{bid level limit}} \{\underbrace{(\text{item, exchange limit, price}), \ldots}_{\textit{Request Set}}\}$$

For each bid, there is a set of three tuples, called the *ask set*, shown on the left hand side of the arrow. This set indicates the items to be sold or given in the exchange. Likewise, the right hand side of the arrow contains a set of three tuples, called the *request set*, which indicates the items to be purchased or taken. Each triple in these sets consists of an item, an *item level upper exchange limit* which is the maximum number of units of the item to be traded (i.e. purchased, sold or exchanged) and a reservation price per unit of the item. There is a logical OR relationship between the items listed in both the ask and request sets. Thus, any subset of items listed in the ask set can be bartered for any subset of items listed in the request set. However, in the MUDAB model, item exchanges are one-to-one so that each unit of an item can be exchanged for one unit of another item. If needed, in order to limit the number of units of items to be traded for the bid, the bid may also have a *bid level upper exchange limit* which is shown immediately above the arrow. This limit value puts an upper bound on the number of items to be traded for the corresponding bid.

For a barter request, in addition to the items to be bartered, the bidder should also declare a price difference between these items if the reservation prices of the items are different. In the previous DAB model, this price difference is directly encoded inside the bids along with the items to be bartered. This scheme is suitable for the previous model since its bidding language does not allow bidders to declare more than one item in both sides of a bid. However, in the MUDAB model, the bidding language allows both the ask and request sets to contain more than one item. In this case, a differential price is required for each pair of the items to be bartered. For instance, assume that a bidder wants to exchange items A, B, C with items D, E, F. For this request he may declare the differential prices between these items as shown in Figure 4.1. In this scheme, the number of prices to be declared is equal to the product of the size of the ask and the size of the request set. Since this number gets larger as the sizes of these sets increase, in the MUDAB model, instead of declaring differential prices, each bidder declares prices for the items they want to give and they want to take as shown in Figure 4.2. The price differences between the declared prices of items reflect the differential prices. This scheme reduces the number of prices to be declared for a bid to the sum of the sizes of the ask and request sets. However, it should be noted that prices of the items declared by the bidders inside the bids are totally subjective and their actual values affect neither the satisfiability of the bids nor the amount money to be paid/earned by the bidders as long as the price differences between the items reflect the additional (differential) amounts that the bidders are willing to give or get. To support the realism of this bidding format, we interviewed used car sellers. When they barter cars, they in fact go through the process of estimating the prices of the car to be given and to be obtained. Then, they determine the price difference which they will offer as additional payment to be received or given. This real life observation is another motivation for using this pricing scheme.

The following example illustrates the details of the MUDAB model and the new bidding language. Assume that four different bidders participate in a market. At the beginning, Bidder 1 owns 50 units of item A and 30 units of item B. Bidder 2 owns 40 units of item C and 20 units of item D. Bidder 3 owns 20 units of item D and Bidder 4 does not have any item.

Figure 4.3 illustrates the example scenario and its mathematical representation using the new bidding language is given in Figure 4.4. Bid 5 is an example of a basic *barter bid*. In this bid, Bidder 3 wants to exchange *up to* 20 units of item D that he owns for the same number of units of item A. The bidder considers the price of each unit of item D as \$60 and the price of each unit of item A as \$120. Assuming that this bid is satisfied, meaning x units of item D are exchanged for x units of item A

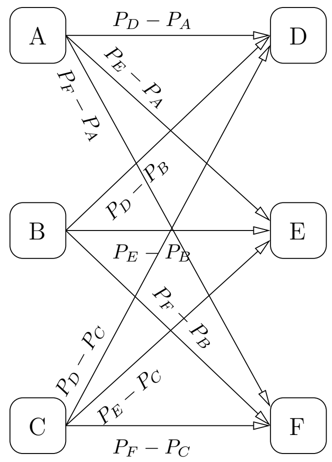


Figure 4.1. Scheme of expressing differential prices for each pair of items. P_X denotes the price of the item X. This scheme requires $|Ask\ Set| * |Request\ Set| = 9$ prices to be declared.

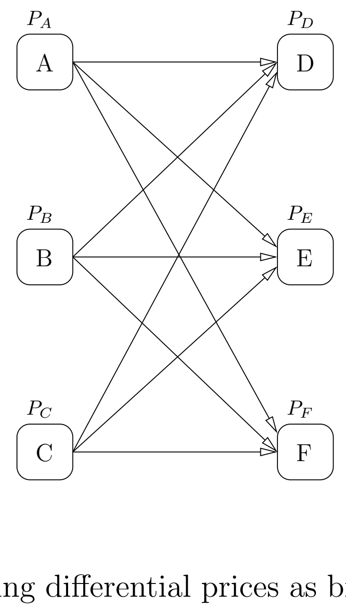


Figure 4.2. Scheme of expressing differential prices as bidder specified prices on items. P_X denotes the price of the item X. This scheme requires $|Ask\ Set| + |Request\ Set| = 6$ prices to be declared.

where $1 \leq x \leq 20$, the amount of money that the bidder must pay for this bid, which is called the *bid payoff*, is $x \cdot \$(120 - 60 = 60)$. If the reservation price of item A was lower than that of item D, then the bid payoff would be negative meaning that the bidder would earn this amount of money if the bid was satisfied. Since there is only one tuple in both the exchange and the request sets, the upper limits for items D and A are unnecessary as long as they are equal or higher than the bid level upper exchange limit. However, these limits are included in order to keep the integrity of the representation of the bidding language. If an upper limit value is not necessary, and therefore not declared, then it can be simply represented with the positive infinity (or the largest positive number).

In order to support sale and purchase requests inside the bids, a special item named $MONEY$ is introduced in this model. For instance, bid 4 is an example of a *sale bid*. In this bid, Bidder 2 wants to sell up to 20 units of item C. Since the model enforces one-to-one exchanges, the bidder declares that he wants to exchange up to 20 units of item C with the same number of units of item $MONEY$. Since the reservation price of item $MONEY$ is \$0, and assuming that 20 units are sold, the bid payoff would be $\$-4,800$ $(= 20 \cdot (0 - 240))$ meaning that the bidder would earn \$4,800. In the MUDAB model, each unit of an item to be sold should be exchanged for one unit of item $MONEY$, and likewise one unit of item $MONEY$ should be exchanged for each unit of an item to be purchased. The item level upper exchange limit for item $MONEY$ limits the number of units of item to be sold or purchased for that bid. Unless a bidder requests a further restriction for the total number of units of items to be purchased or sold in all of his bids, each bidder is assigned with an infinite number of units of item $MONEY$.

Bid 6 is an example of a *purchase bid*. In this bid, Bidder 4 declares that he wants to purchase up to 30 units of item B and up to 20 units of item D but no more than 40 units of both items which is indicated by the bid level upper exchange limit.

Bid 3 is a more complicated example in which the bidder wants to exchange two different items C, D for items A, B. The upper limits on the number of units to be

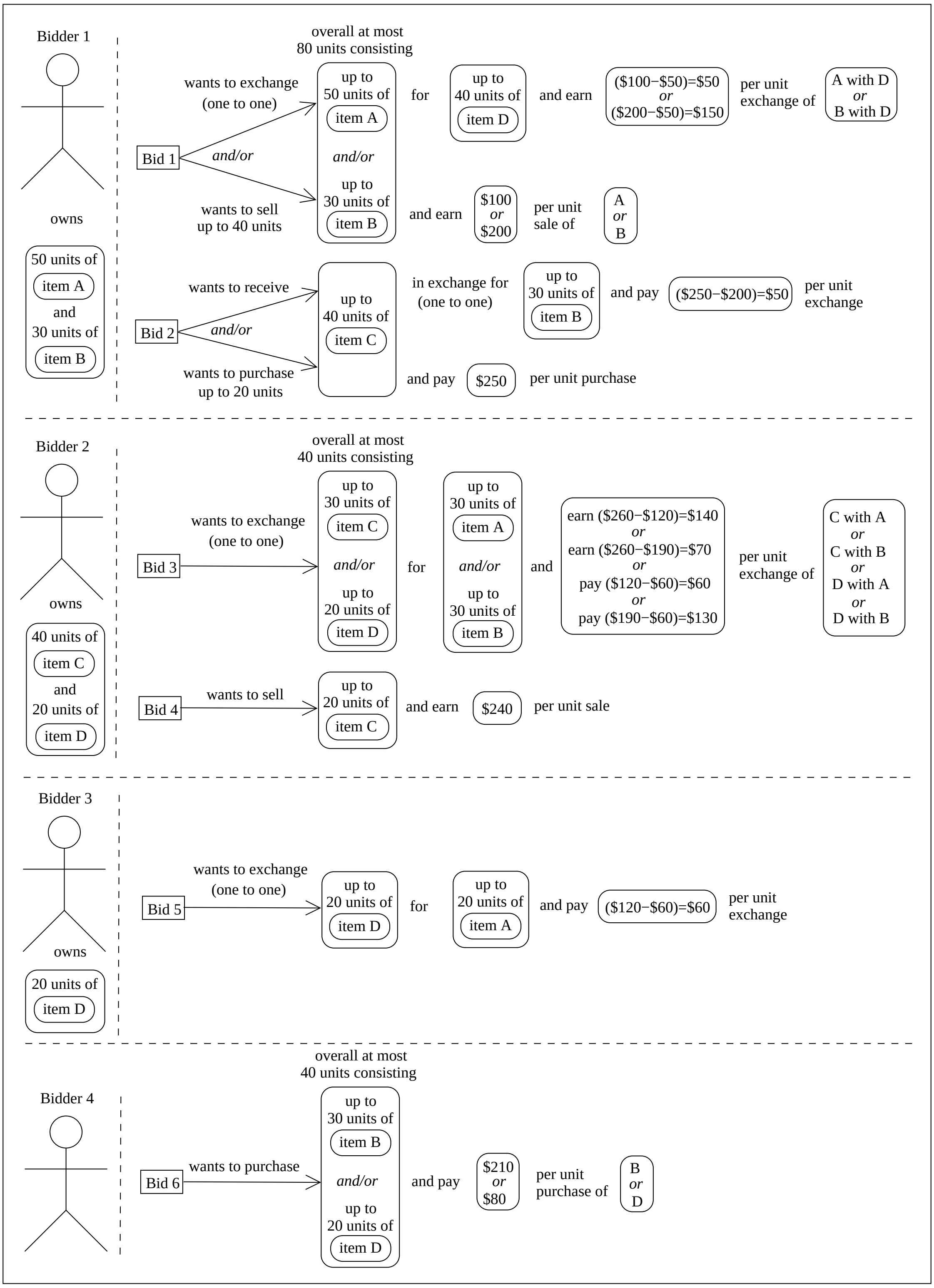


Figure 4.3. Example market scenario for illustrating the MUDAB model. Note that the model prevents bidders to give more than the items they own even if they bid so.

a) Ownership Table:

	Item A	Item B	Item C	Item D
Bidder 1	50 units	30 units	-	-
Bidder 2	-	-	40 units	20 units
Bidder 3	-	-	-	20 units
Bidder 4	-	-	-	-

b) Bids:

Bid Representation:

$$\underbrace{\{(\text{item, exchange limit, price}), \ldots\}}_{Ask\ Set} \xrightarrow[\text{limit}]{\text{bid level}} \underbrace{\{(\text{item, exchange limit, price}), \ldots\}}_{Request\ Set}$$

Bidder 1

Bid 1 :$\{(A,50,\$100),(B,30,\$200)\} \xrightarrow{80} \{(D,40,\$50),(MONEY,40,\$0)\}$

Bid 2 :$\{(B,30,\$200),(MONEY,20,\$0)\} \xrightarrow{40} \{(C,40,\$250)\}$

Bidder 2

Bid 3 :$\{(C,30,\$260),(D,20,\$60)\} \xrightarrow{40} \{(A,30,\$120),(B,30,\$190)\}$

Bid 4 :$\{(C,20,\$240)\} \xrightarrow{20} \{(MONEY,20,\$0)\}$

Bidder 3

Bid 5 :$\{(D,20,\$60)\} \xrightarrow{20} \{(A,20,\$120)\}$

Bidder 4

Bid 6 :$\{(MONEY,40,\$0)\} \xrightarrow{40} \{(B,30,\$210),(D,20,\$80)\}$

Figure 4.4. Representation of the example market scenario in Figure 4.3. The scenario is represented using the bidding language of our model.

exchanged are 30, 20, 30 and 30 respectively. The maximum number of units allowed for the exchange is 40. Suppose that this bid is satisfied, and let x_C, x_D, x_A and x_B be the numbers of units of items C, D, A and B that are exchanged ($0 \leq x_C \leq 30$, $0 \leq x_D \leq 20$, $0 \leq x_A \leq 30$ and $0 \leq x_B \leq 30$). Then, the payoff of this bid is equal to $(x_A \cdot \$120 + x_B \cdot \$190) - (x_C \cdot \$260 + x_D \cdot \$60)$. It should be noted that the total number of items to be given should be equal to the total number of items to be taken, that is $(x_C + x_D = x_A + x_B \leq 40)$.

By introducing the special item $MONEY$, the model allows bidders to express more complex preferences of purchase, sell and exchange inside a single bid. For instance, in the first bid, the bidder wants to exchange up to 40 units of his inventory for the same number of units of item D and at the same time he wants to sell the rest of his inventory without differentiating which item in his inventory is exchanged or sold. The outcome of this bid can be such that the bidder exchanges 40 units of item A for 40 units of item D, and sells 10 units of item A and 30 units of item B. However, the outcome can also be that the bidder exchanges 10 units of item A and 30 units of item B for 40 units of item D, and sells 40 units of item A. In both cases the amount of money that the bidder earns, that is \$9,000, would not be different. In bid 2, on the other hand, the bidder wants to have up to 40 units of item C without differentiating the cases in which "he exchanges 30 units of item B for 30 units of item C, and purchases 10 units of item C", or "he exchanges 20 units of item B for 20 units of item C, and purchases 20 units of item C". The exchange limits in this bid also puts a limit on the amount of money that the bidder would have to pay which is \$(20 * 250 + 20 * 50 = 6,000). It should be noted that although Bidder 1 offers to give 30 units of item B in both his bids, bid 1 and 2, the model does not allow the bidder to give more than the number of units of items he owns in the final outcome.

In the given example, the reservation price of the item $MONEY$ is set to zero for the ease of understanding. However, the model also allows assigning reservation prices for the item $MONEY$ in order to increase the expressibility of the bids. For instance, assume that a bidder owns 10 instances of item A and requests 10 instances of item B. He considers the price difference between each unit of these items as \$100.

Furthermore, if bartering would not be possible, then he wants to sell each unit of his item A for the price of \$80 or purchase each unit of item B for the price of \$160. This preference of the bidder can be expressed by the following bid:

$$\{(A, 10, \$100), (MONEY, 10, \$40)\} \xrightarrow{10} \{(B, 10, \$200), (MONEY, 10, \$20)\}$$

Here, it should be noted that if reservation prices other than zero are assigned to the item $MONEY$, then the reservation price of the item $MONEY$ in the ask set cannot be lower than the reservation price of the item $MONEY$ in the request set. Otherwise, this situation may cause the bidder to pay money without getting any item.

4.5. Formulation of the MUDAB Model

The MUDAB model is formally defined as follows: Let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of m bidders and $R = \{MONEY, r_2, r_3, \ \ldots, r_n\}$ be the set of n items in the market. The first item $MONEY(= r_1)$ represents the special item which is used for purchase and sale requests inside the bids. Let O be an m-by-n ownership matrix where element o_{ij} defines the number of units of item r_j owned by bidder c_i ($o_{ij} \in \mathbb{Z}^+ \cup 0$). We define B_i as the set of bids submitted by the bidder c_i and the set of all bids, B, is defined as $B = \bigcup_{i=1}^{m} B_i$. Each bid b_k is denoted by a triplet, $b_k = (A_k, u_k, R_k)$, where A_k is the ask set, u_k is the bid level upper exchange limit ($u_k \in \mathbb{Z}^+$) and R_k is the request set of the bid b_k. The ask set is used for declaring items to be bartered or sold in the market. It consists of y three tuples, $A_k = \{(i'_1, u'_1, p'_1), \ldots, (i'_y, u'_y, p'_y)\}, \ y \in \mathbb{Z}^+$, and in each tuple (i'_t, u'_t, p'_t), i'_t denotes the index of the item to be bartered or sold, u'_t denotes the item level upper exchange limit, and p'_t is the reservation price of the item $(1 \leq t \leq y, r_{i'_t} \in R, u'_t \in \mathbb{Z}^+, p'_t \in \mathbb{R})$. The request set, on the other hand, is used for declaring items to be bartered for or purchased. It consists of z three tuples, $R_k = \{(i''_1, u''_1, p''_1), \ldots, (i''_z, u''_z, p''_z)\}, \ z \in \mathbb{Z}^+$, and in each tuple (i''_v, u''_v, p''_v), i''_v denotes the index of the item to be taken or purchased, u''_v denotes the item level upper exchange limit, and p''_v is the reservation price of the item $(1 \leq v \leq z, r_{i''_v} \in R, u''_v \in \mathbb{Z}^+, p''_v \in \mathbb{R})$. In this definition, in order to prevent the confusion and increase the readability, the superscript ($'$) is used for the variables of the ask set and the superscript ($''$) is used for the variables of the request set.

A bid b_k is called *satisfiable* if and only if the following equalities hold:

$$x'_{kt} \leq u'_t \ (\forall t \,|\, (i'_t, u'_t, p'_t) \in A_k),$$
$$x''_{kv} \leq u''_v \ (\forall v \,|\, (i''_v, u''_v, p''_v) \in R_k)$$
$$1 \leq \sum_{\forall t \,|\, (i'_t, u'_t, p'_t) \in A_k} x'_{kt} = \sum_{\forall v \,|\, (i''_v, u''_v, p''_v) \in R_k} x''_{kv} \leq u_k.$$

where x'_{kt} denotes the number of units of item to be given by tth tuple of the ask set A_k of the bid b_k, and x''_{kv} denotes the number of units of items to be taken by vth tuple of the request set R_k of the bid b_k. The bid payoff (p_k) for the bid b_k is calculated as follows:

$$p_k = \sum_{\forall v \,|\, (i''_v, u''_v, p''_v) \in R_k} p''_v x''_{kv} - \sum_{\forall t \,|\, (i'_t, u'_t, p'_t) \in A_k} p'_t x'_{kt}$$

The *winner determination problem of the MUDAB model* is defined as finding a set of mutually satisfiable bids such that the sum of the payoffs of the bids is maximized. If the maximum sum of the bid payoffs is negative for a problem instance, then this means that there is no feasible solution unless the deficit is subsidized by the auctioneer.

It should be noted that the number of units of the item $MONEY$ owned by a bidder c_i, that is o_{i1}, defines an upper bound for the difference between the number of units of item that he can purchase and the number of units of item that he can sell in all of his bids. If such a restriction is not necessary for the bidder, then this value is set to the largest positive number.

The linear integer programming formulation of the winner determination problem is as follows:

$$\text{maximize} \sum_{\forall k \,|\, b_k \in B} \left(\sum_{\forall v \,|\, (i''_v, u''_v, p''_v) \in R_k} p''_v x''_{kv} - \sum_{\forall t \,|\, (i'_t, u'_t, p'_t) \in A_k} p'_t x'_{kt} \right) \tag{4.1}$$

$$\text{subject to} \sum_{\forall v \,|\, (i''_v, u''_v, p''_v) \in R_k} x''_{kv} - \sum_{\forall t \,|\, (i'_t, u'_t, p'_t) \in A_k} x'_{kt} = 0 \quad (\forall k \,|\, b_k \in B) \tag{4.2}$$

$$\sum_{\forall k, v \,|\, b_k \in B \,\wedge\, (i''_v, u''_v, p''_v) \in R_k \,\wedge\, i''_v = j} x''_{kv} - \sum_{\forall k, t \,|\, b_k \in B \,\wedge\, (i'_t, u'_t, p'_t) \in A_k \,\wedge\, i'_t = j} x'_{kt} = 0 \quad (\forall j \,|\, r_j \in R) \tag{4.3}$$

$$\sum_{\forall k, t \,|\, b_k \in B_i \,\wedge\, (i'_t, u'_t, p'_t) \in A_k \,\wedge\, i'_t = j} x'_{kt} \leq o_{ij} \quad (\forall i, j \,|\, c_i \in C, r_j \in R) \tag{4.4}$$

$$x'_{kt} \leq u'_t \quad (\forall k, t \,|\, b_k \in B \,\wedge\, (i'_t, u'_t, p'_t) \in A_k) \tag{4.5}$$

$$x''_{kv} \leq u''_v \quad (\forall k, v \,|\, b_k \in B \,\wedge\, (i''_v, u''_v, p''_v) \in R_k) \tag{4.6}$$

$$\sum_{\forall t \,|\, (i'_t, u'_t, p'_t) \in A_k} x'_{kt} \leq u_k \quad (\forall k \,|\, b_k \in B) \tag{4.7}$$

$$x'_{kt}, x''_{kv} \in \mathbb{N} \quad (\forall k, t, v) \tag{4.8}$$

In this formulation, the objective line in Eq.(4.1) maximizes the sum of the payoffs of all satisfied bids. Eq.(4.2) satisfies the one-to-one exchange requirement of the model such that for every bid the number of units of items to be taken must be equal to the number of units of items to be given. Eq.(4.3) imposes a basic feature of any market. For any bidder, in order to purchase or get an item in the exchange, there must be a bidder that sells or gives the same item. Therefore, for each item, the number of units of items taken by all bidders must be equal to the number of units of items given by all bidders. Furthermore, it is clear that for each item, no bidder can give more than the number of units he owns and this restriction is enforced with Eq.(4.4). Eq.(4.5) and Eq.(4.6) apply the item level upper bound restrictions and finally Eq.(4.7) ensures that for any bid, the number of units of items to be exchanged does not exceed the bid level upper exchange limit.

4.5.1. An Issue Related to the Unrequested Items

In the MUDAB model, bidders should work out and declare the price differences between the items they want to give and the items they want to get inside the bids. Since the reservation prices of the items are subjective, the following situation may occur. Consider the following bids:

$$\begin{aligned}
&\text{Bidder 1: } b_1 = \{(A, 1, \$230)\} \xrightarrow{1} \{(MONEY, 1, \$0)\} \\
&\text{Bidder 2: } b_2 = \{(B, 1, \$350)\} \xrightarrow{1} \{(A, 1, \$220)\} \\
&\text{Bidder 3: } b_3 = \{(C, 1, \$60)\} \xrightarrow{1} \{(B, 1, \$380)\} \\
&\text{Bidder 4: } b_4 = \{(D, 1, \$20)\} \xrightarrow{1} \{(C, 1, \$70)\}
\end{aligned}$$

According to the model definition, these four bids cannot be satisfied together since there is no bid that requests item D. However, in fact these bids are satisfiable since the sum of the bid payoffs $(220 + 380 + 70) - (230 + 350 + 60 + 20) = 10$ is greater than zero and all the bidders get the items or the amount of money they want. This issue can be resolved by introducing a dummy bid (b_d) in order to buy *all* units of *all* items for the price zero:

$$b_d = \{(MONEY, u_d, \$0)\} \xrightarrow{u_d} \{(r_2, u''_2, \$0), (r_3, u''_3, \$0), \ldots, (r_n, u''_n, \$0)\}$$

where u''_j is the number of available units of item r_j ($u''_j = \sum_{i=1}^{m} o_{ij}$ $(j = 2, 3, \ldots, n)$) and u_d is the number of available units of all items in the market ($u_d = \sum_{i=1}^{m} \sum_{j=2}^{n} o_{ij}$). This dummy bid can be automatically entered by the auctioneer after all the bids have been collected from the bidders. After solving the MUDAB problem, either the auctioneer could keep the profit as well as the unrequested items given in the winning bids (item D in the above example) or alternatively let the owners of the winning bids with the unrequested items get what they want without giving any item (e.g. in the above example, Bidder 4 could be allowed to get item C without giving item D).

4.6. Solution Procedure

In this section, we will introduce an algorithm for transforming the winner determination problem of the MUDAB model to a minimum cost circulation flow problem [106]. Let $N(V, A, l, u, c, b)$ denote a network with node set V, arc set A, lower bound $l(v, w)$, capacity $u(v, w)$, cost $c(v, w)$ for each arc $(v, w) \in A$, and supply/demand value $b(v)$ for each node $v \in V$. The minimum cost circulation network can be constructed as follows:

- The set of nodes, V, consists of three types of nodes:
  - (i) Item nodes r_j for each item $r_j \in R$ including the special item $MONEY$.
  - (ii) Ownership nodes o_{ij} for each bidder $c_i \in C$ and for each item $r_j \in R$ if $o_{ij} > 0$.
  - (iii) Split bid nodes b'_k and b''_k for each bid $b_k \in B$.
- The set of arcs, A, consists of four types of arcs:
  - (i) An arc $<r_j, o_{ij}>$ for each bidder $c_i \in C$ and for each item $r_j \in R$ if $o_{ij} > 0$. The capacity of the arc is equal to o_{ij} and the cost is equal to 0.
  - (ii) An arc $<o_{ij}, b'_k>$ for each bidder $c_i \in C$, for each item $r_j \in R$ and for each bid $b_k = (A_k, u_k, R_k) \in B_i$ if there exists a t such that $(i'_t, u'_t, p'_t) \in A_k$ and $i'_t = j$. The capacity is equal to u'_t and the cost is equal to p'_t.
  - (iii) An arc $<b'_k, b''_k>$ for each bid $b_k = (A_k, u_k, R_k) \in B$ with the capacity equal to u_k and the cost equal to 0.
  - (iv) An arc $<b''_k, r_j>$ for each bid $b_k = (A_k, u_k, R_k) \in B$ and for each item $r_j \in R$ if there exists a v such that $(i''_v, u''_v, p''_v) \in R_k$ and $i''_v = j$. The capacity of the arc is equal to u''_v and the cost is equal to $-p''_v$.

There is neither supply nor demand for any node in the network, and therefore $b(v) = 0$ for every node $v \in V$. Also, lower bounds $l(v, w)$ for all arcs $(v, w) \in A$ are set to 0. The minimum cost circulation network for the example problem given in Section 4.4 and its solution can be seen in Figures 4.5 and 4.6 respectively.

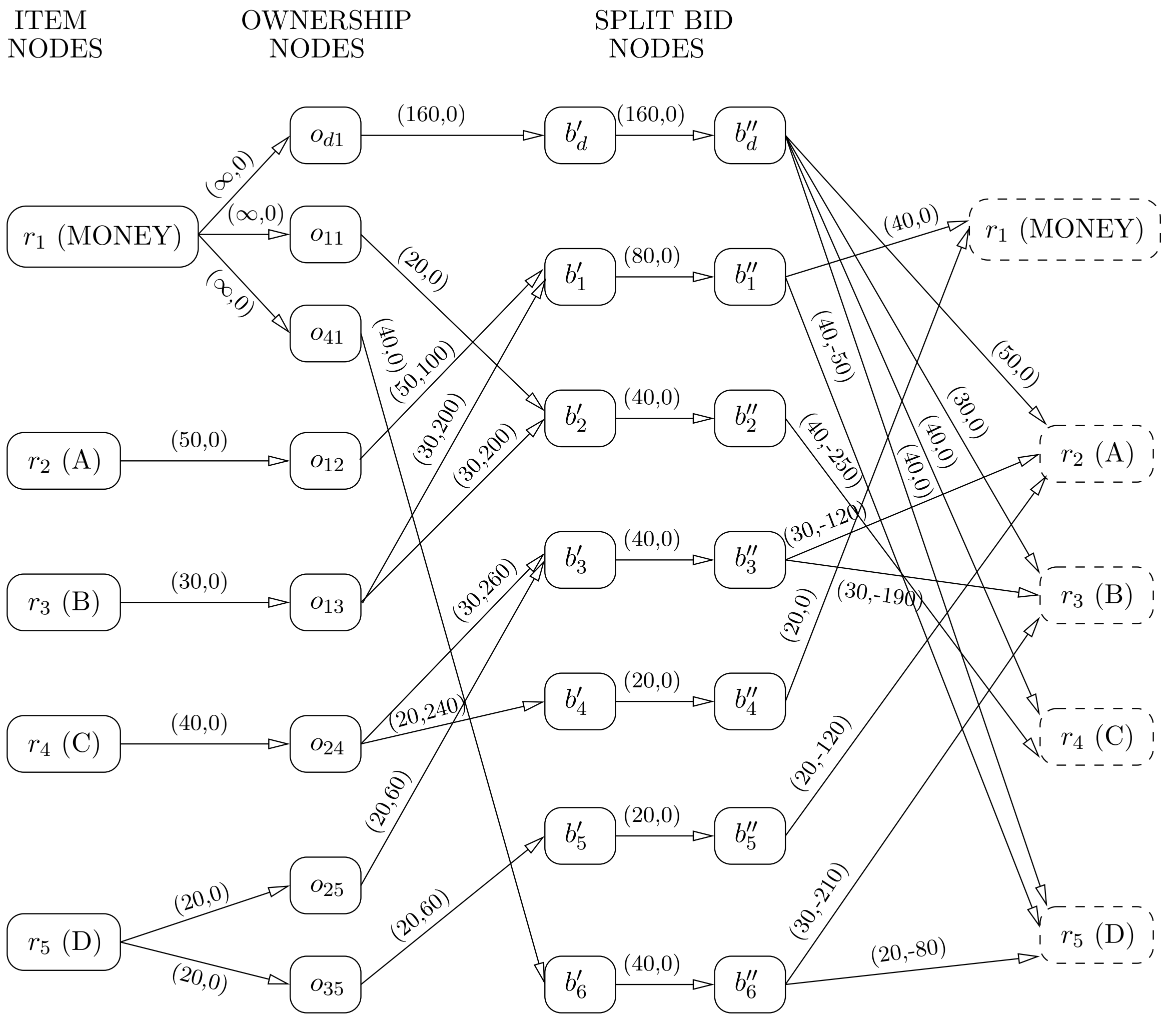
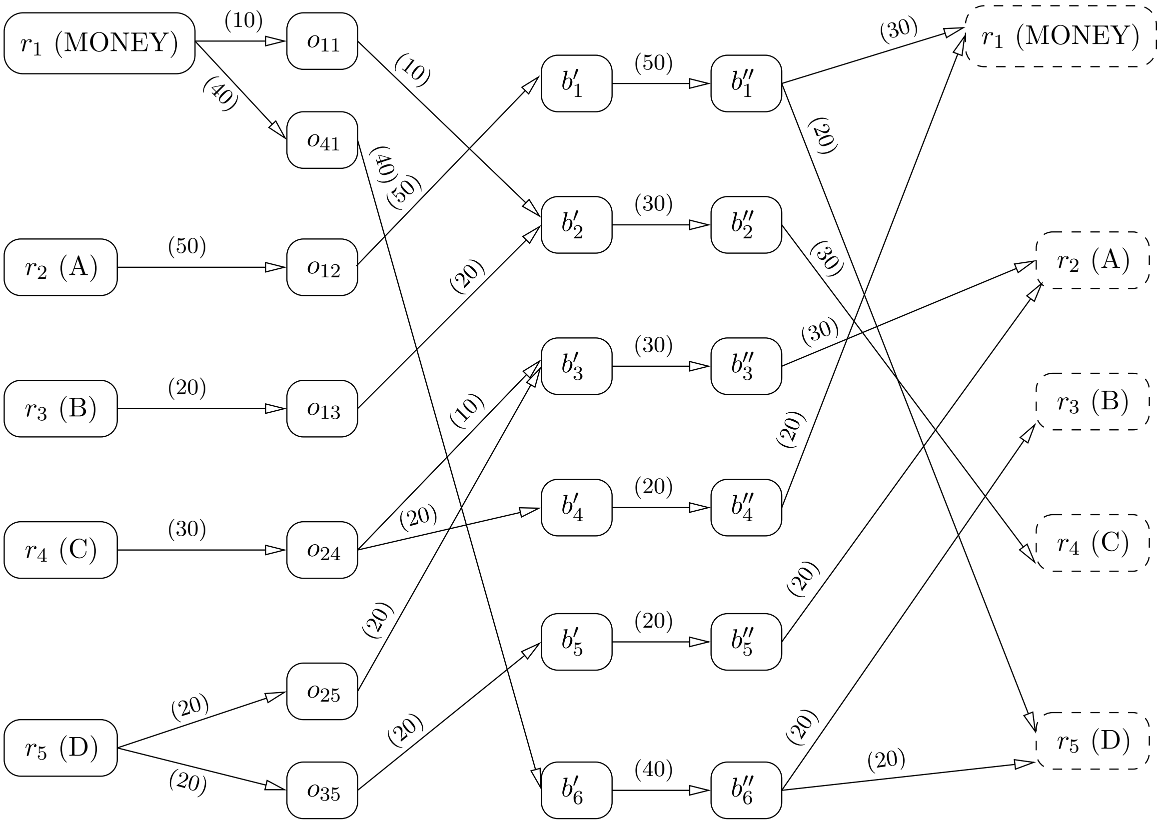


Figure 4.5. Minimum cost circulation network of the example in Section 4.4. (capacity, cost) values are presented on the arcs. Note that the dashed nodes located on the right side are the same nodes located on the left side.

The correctness of the network can be verified as follows. The integer variables x'_{kt} are represented with the arcs of type (ii) and x''_{kv} are represented with the arcs of type (iv). The costs of the type (ii) arcs are the reservation prices of the items declared in the ask set whereas the costs of the type (iv) arcs are the additive inverses of the reservation prices of the items declared in the request set. Since the costs of the other arcs are zero, finding the minimum cost circulation flow means optimizing the objective function in Eq.(4.1). The constraints in Eq.(4.2) and Eq.(4.3) are applied by the flow conservation property of the network for the nodes of type (iii) and the nodes of type



Bid 1: Bidder 1 exchanges 20 units of item A for 20 units of item D, and sells 30 units of item A. He earns \$4000.

Bid 2: Bidder 1 exchanges 20 units of item B for 20 units of item C, and buys 10 units of item C. He pays \$3500.

Bid 3: Bidder 2 exchanges 10 units of item C and 20 units of item D for 30 units of item A. He earns \$200.

Bid 4: Bidder 2 sells 20 units of item C. He earns \$4800.

Bid 5: Bidder 3 exchanges 20 units of item D for 20 units of item A. He pays \$1200.

Bid 6: Bidder 4 buys 20 units of item B and 20 units of item D. He pays \$5800.

Figure 4.6. Solution of the minimum cost circulation network in Figure 4.5. (flow) values are presented on the arcs.

(i) respectively. The upper limits on the arcs of type (i) ensure that no bidder can give more than the number of units of items he owns which is stated in Eq.(4.4). Finally, the upper limits on the arcs of type (ii) and (iv) apply the item-level upper exchange limit constraints and the upper limits on the arcs of type (iii) apply the bid-level upper exchange limit constraints.

4.7. Experimental Results

We have prepared test cases for observing the performance and the scalability of the MUDAB model in practice. For solving these cases, we use CS2 software which contains an implementation of scaling push-relabel algorithm for the minimum cost flow problems [109, 110]. Tests are conducted on a dedicated 64 bit Intel Xeon 2.66 GHz workstation with eight gigabytes of memory using single core.

The test cases are generated using 12 different configurations for simulating different market sizes. In these configurations, the number of bidders varies between 10,000 and 100,000, and the number of bids varies between 100,000 and 1,000,000. The sizes of the request sets (excluding the special item MONEY) are drawn uniformly from the intervals $[0, 2]$, $[0, 10]$ and $[0, 20]$ which are denoted as *small*, *medium* and *large* sizes respectively. Each bidder is assumed to have an average of five different items and up to 100 units of each of these items. The maximum size of an ask set is limited to the number of items of the bid owner. The probability of the bidder to include the special item MONEY inside the ask or request sets is set to 0.25. The rest of the parameters of the test case generator, the generated test cases and their solutions can be obtained from [130].

The results of the experiment are presented in Table 4.5. For each of the 12 configurations, 20 test cases are generated and the average number of items, the average number of nodes and edges in the network, the average execution time of the network solver and the corresponding standard deviation are reported. The optimum solution to the problem instances containing up to 200,000 bids can be solved within a minute. Even the larger instances with 1,000,000 bids and the network size of approximately 3,000,000 nodes and 15,000,000 edges are solvable in less than five minutes demonstrating the scalability of the model in practice.

Table 4.5. Average network sizes and execution times of the network solver for test cases. Each test case configuration consists of 20 instances.

Bids	*Request Set Size*	*Nodes*	*Edges*	*Execution Time (seconds)* *mean*	*stdev*
100,000	small	310,086	639,944	9.19	0.58
	medium	310,039	1,018,887	12.36	0.61
	large	310,017	1,514,825	15.99	0.82
200,000	small	619,956	1,278,457	21.99	1.39
	medium	620,099	2,038,985	29.77	1.85
	large	620,006	3,029,529	38.47	1.41
500,000	small	1,549,931	3,196,936	64.15	4.15
	medium	1,549,876	5,092,334	83.57	4.97
	large	1,550,019	7,573,373	108.39	3.21
1,000,000	small	3,099,875	6,393,790	137.03	10.80
	medium	3,100,030	10,188,917	193.61	6.05
	large	3,100,136	15,150,880	247.84	14.12

4.8. Conclusion

In this chapter, we contribute a market model and an algorithm that improve DA based e-markets by allowing barter bids. The MUDAB model allows traders to declare both direct and differential barter requests along with the usual sale and purchase requests. It also extends the previous DAB model which allows bartering of only single items. The MUDAB model can be used virtually in every market in which DA rules are applicable. Furthermore, the MUDAB model utilizes a more powerful bidding language which improves the market outcome when compared with the DAB model. Since the MUDAB model is the superset of both multi-unit DA and the DAB models, the allocative efficiency of the MUDAB model is also guaranteed to be better or at least as good as that of these models. Even with these additional features, the MUDAB market can be cleared fast using the proposed polynomial-time network flow based algorithm. Test instances involving millions of bids can be solved within minutes which opens the way for building realistic e-markets for massive number of users. A drawback of our model and the corresponding algorithm is that it does not allow package bidding, and therefore, the item exchanges are always one-to-one. Even though our model can be generalized to many-to-many exchange option, this will make the problem computationally harder, i.e. NP-Hard, as proven in [81].

The implementation of the model is also straightforward; bids are collected from the bidders within a predefined time period. Then, the optimum allocation is found by running our algorithm and the results are published. Unsatisfied bids could be transferred to the next bidding round unless they are withdrawn. The length of the rounds can be determined according to the rate of submission of bids. For instance, for the described e-media market the length of a round could be as short as 30 minutes whereas for a property exchange the length could be as long as 30 days.

The benefits of bartering mechanism have already been demonstrated in real life. For instance, in the United States, there are approximately 400 barter exchange companies and it is estimated that over 350,000 businesses are involved in bartering activities. According to statistics published by The International Reciprocal Trade Association, $8.25 billion worth of goods and services were traded through barter exchange companies in 2004 [131]. Besides the barter exchanges, there are also non-profit and commercial online platforms that provide bartering services. For instance, swaptree.com offers their members to barter books, music, DVDs or video games. The method of bartering used is the direct bartering without involvement of money. Similarly, rehashclothes.com provides a platform for bartering clothing and goswap.org does the same for bartering houses and land. An example for non-profit platforms would be textswap.com which provides textbook exchange services for college students.

These examples motivate us to envision future e-markets incorporating the differential barter mechanism for increasing the throughput of the real life markets. The known difficulty of finding the double coincidence of wants in bartering is resolved by the algorithmic engines that facilitate multilateral (multi-way) bartering among two or more parties. Furthermore, we believe that the introduced features of the model and the web-based access to the markets will also ease the transition from otherwise producer-consumer oriented markets to the markets in which consumers become traders who buy and sell commodities in order to make profit just like in eBay. This transition will enrich the markets greatly as it has been seen in stock exchanges.

5. DOUBLE AUCTION WITH LIMITED COVER MONEY MODEL

5.1. Introduction

Secondary markets in which used durable goods are exchanged play an important role in overall economic activity with multi-billion dollars of transaction volumes. For instance, in 2009 approximately $80.5 billion worth of used vehicles were sold in NAAA's North American member auctions according to the association's 13th Annual Survey [128]. Other examples from the U.S. are the used book market with approximately $2.2 billion purchases in 2004 [132] and the used video game market with more than $2 billion purchases [133] in 2009. Considerable amount of these trading volumes has been carried out through online markets. Again, for instance, approximately 21% of the sales conducted by NAAA's North American members are via Internet based sources. This ratio is much higher for used book market such that online booksellers are responsible for two-thirds of the general interest used book sales [132].

Secondary markets are commonly considered as a threat for suppliers as it is thought that used good sales cannibalize new good sales. Although, some positive evidences exist for specific markets, for instance Smith et al. [134] find that 86% of the used DVD sales through Amazon.com cannibalize the new DVD sales, this cannot be generalized. The reason is that DVDs are digital media which are less prone to loss in quality when used, their utilities are dramatically diminished after they are seen, and furthermore, they can easily be copied using specialized software. The cannibalization rates of used books and CDs sold through Amazon.com, in contrast, are quite low with 16% [135] and 24% [134], respectively. It is shown that unless the market is a monopoly or a supplier has very high share in the market, secondary markets can be beneficial for suppliers since the presence of these markets increase the profitability of new goods yielding higher supplier welfare [136, 137]. Secondary markets have also an indirect benefit for suppliers. In his open letter to Amazon.com's used booksellers dated April 14, 2002, Jeff Bezos, CEO of Amazon.com, indicated this benefit as *"... when a customer sells used books, it gives them a budget to buy more new books."*. From the buyers' perspective on the other hand, the presence of secondary markets are always optimal increasing buyers' welfare [136]. These benefits also provide incentive for intermediaries to operate such markets, the well-known of examples of which are eBay and Amazon.com. Ghose et al. [135] estimate an additional profit of approximately $65 million for Amazon.com from used book sales with an estimated increase of approximately $88 million in the total social welfare.

In this chapter, we propose an e-market model called double auction with limited cover money (DALCOM) model which is designed especially for secondary markets. The model utilizes the discrete-time DA institution, that is the clearinghouse, for the trading of used goods as well as new ones. Market participants can have both buyer and seller roles, and therefore each bidder can put forward items for sale as well as place bids for purchase. The model extends the DA institution such that it allows bidders to put a limit on their budget by providing a mechanism for declaration of an amount of cover money. The model ensures that for each bidder what is spent on purchased items minus the proceeds of sold items do not exceed the declared cover money amount. This mechanism encourages bidders to participate in e-markets without a risk of having a budget deficit. Besides, a bidder may also be indifferent to multiple items, for instance there may be multiple sellers of the same textbook or the bidder may be interested in a specific set of novels in a book market. The DALCOM model further provides a mechanism for handling such situations so that in their purchase bids, the bidders are allowed to declare a list of items and put a limit on the number of items to be purchased in that list. Thus, this mechanism enables bidders to express their preferences of substitutabilities in the market which yields in a better allocation of resources. For pricing the items, the well-known k-DA policy is used. The winner determination problem of the model is formally defined and formulated using linear integer programming. It is proven that the decision version of the winner determination problem is NP-complete and even it is inapproximable unless $P = NP$. We have also designed a test case generator with wide-range of configurable parameters with which the performance of the CPLEX MIP solver for this problem is demonstrated on a comprehensive test suite.

In the next section, the DALCOM model is explained in detail on a sample scenario from the used book market. In Section 5.3, the model is formally defined, the winner determination problem is formulated and the complexity results are presented. The test case generator is explained in Section 5.4 and the experimental results are presented in Section 5.5. Finally, the chapter is concluded in Section 5.6.

5.2. The DALCOM Model

In this section, we introduce the DALCOM model on an example from the used book market. Consider the following scenario which is also illustrated in Figure 5.1. There are five books put up for sale by four bidders. Each bidder declares a reservation price for each book they want to sell which indicates the minimum amount of money the bidder is willing to get if the corresponding book is sold. Along with the reservation prices, each bidder also declares a nonnegative cover money amount which indicates the maximum amount of money that the bidder is willing to spend in addition to the proceeds of the sold items. In this model, each item is considered as unique. This feature allows bidders to differentiate between even the instances of the same items sold by different bidders, since, for instance, the condition of the instance, its warranty status, the reputation of its seller, the location of the item and the associated transfer cost may vary. Note that this does not pose a problem for the bidders who do not differentiate between the identical items since they are allowed declare a list of substitutable items in their bids. In this example, there are five purchase bids submitted by four bidders. In the DALCOM model, a purchase bid contains a request set and an upper purchase limit value and is represented as follows:

$$\langle\ \{\ (\text{Item } A,\ \text{Res. Price for } A),\ (\text{Item } B,\ \text{Res. Price for } B), \ldots\ \},\ \text{Upper Pur. Limit}\ \rangle$$

Items for Sale:

Seller	Item	Reservation Price
Bidder 1:	Book A	\$40
	Book B	\$30
Bidder 2:	Book C	\$25
Bidder 3:	Book D	\$30
Bidder 4:	Book E	\$30

Amounts of Cover Money:

Bidder 1:	\$0
Bidder 2:	\$45
Bidder 3:	\$0
Bidder 4:	\$0
Bidder 5:	\$35

Purchase Bids:

Bidder 1:	b_1:	$\langle \{(\texttt{Book C}, \$35), (\texttt{Book E}, \$35)\}, \texttt{up to 1 item} \rangle$
	b_2:	$\langle \{(\texttt{Book D}, \$40)\}, \texttt{up to 1 item} \rangle$
Bidder 2:	b_3:	$\langle \{(\texttt{Book A}, \$45), (\texttt{Book B}, \$35), (\texttt{Book E}, \$40)\}, \texttt{up to 2 items} \rangle$
Bidder 3:	b_4:	$\langle \{(\texttt{Book A}, \$40), (\texttt{Book C}, \$30), (\texttt{Book E}, \$40)\}, \texttt{up to 1 item} \rangle$
Bidder 5:	b_5:	$\langle \{(\texttt{Book E}, \$35)\}, \texttt{up to 1 item} \rangle$

Figure 5.1. Example problem illustrating sale and purchase bids for DALCOM market.

The request set contains one or more item-reservation price pairs each of which denotes the item a bidder is willing to purchase and the associated reservation price. The reservation price is the maximum amount of money that the bidder is willing to pay for the corresponding item. Similarly, the upper purchase limit denotes the maximum number of items that the bidder is willing to buy among the items listed in the request set. For instance, in bid 5, Bidder 5 declares that he wants to buy book E and he wants to give at most \$35. In bid 3, on the other hand, Bidder 2 wants to buy *at most* two books among books A, B and E and he is willing to pay at most \$45, \$35 and \$40, respectively. Preference of Bidder 1 is more complex than a single bid can express. He wants to purchase book C or E and also book D as long as he stays within his budget. Thus, he submits two bids, bid 1 and 2, for this purpose.

In the DALCOM model, the *utility* of a transaction between a seller and a buyer is defined as the difference between buyer reservation price and the seller reservation price for the item to be exchanged. Negative utility values are not allowed, and therefore bids whose buyer reservation price is smaller than the corresponding seller reservation price are simply discarded. The objective of the model is to maximize the sum of the transaction utilities, and hence enhance the social welfare.

For pricing the items in the exchange, k-DA policy is used and the price of an exchanged item is defined as

$$k \cdot \textit{buyer reservation price} + (1-k) \cdot \textit{seller reservation price}$$

where $k \in [0, 1]$ is the parameter of the k-DA policy which determines how the transaction utility is distributed between the seller and the buyer. For instance, if k is set to 0.5, the utility is equally shared between the parties. In the border cases, $k = 0$ or $k = 1$, the seller or the buyer determines the transaction price alone, respectively. In this study, we set $k = 0.5$ unless otherwise noted for the numerical examples without losing generality.

The advantage of introducing cover money feature can be seen on the example. If this auction was conducted using the traditional double auction mechanism, only Bidder 2 and Bidder 5 could submit a purchase bid, since the budgets of the other bidders do not permit them to purchase new books without selling their old books first. In this case, only the third pair in bid 3 would be satisfied, that is Bidder 2 would purchase book E from Bidder 4 while paying $0.5 \cdot \$40 + 0.5 \cdot \$30 = \$35$. The total utility would be \$40 - \$30=\$10 and the total transaction volume would be \$35. However, in the DALCOM model, all the listed bids can be submitted and four of them, bids 1-4 are satisfied simultaneously. The outcome is as follows: Bidder 1 buys books C and D, Bidder 2 buys books A and B and Bidder 3 buys book E. In this case, the total utility increases to \$40 and the total transaction volume increases to \$175. Furthermore, these transactions do not cause any budget deficit for the bidders.

The implementation of the model is also straightforward as in the case of the previous models. Within a predefined time period, sealed bids are collected from the bidders and after the end of the period the market is cleared by solving the optimization problem which is introduced in the next section. Unsatisfied bids could be transferred to the next bidding round according to the preferences of the bidders. The length of the rounds can be determined according to the market to which the model is applied and the rate of submission of bids in the market. The longer periods result in better allocative efficiency but they also cause less trading volume to occur per unit time.

5.3. Formulation of the DALCOM Model

The DALCOM model is formally defined as follows: Let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of m bidders and T_i be the set of items to be sold by bidder c_i $(1 \leq i \leq m)$. The set of all items, $T = \{t_1, t_2, \ldots, t_n\}$, is defined as $T = \bigcup_{i=1}^{m} T_i$ $(\forall i, i' \,|\, T_i \cap T'_i = \emptyset)$. Let $P = (p_{t_1}, p_{t_2}, \ldots, p_{t_n})$ be the tuple of reservation prices of items where p_{t_j} is the owner-determined reservation price of the item t_j that indicates the minimum amount for which the owner of the item t_j is willing to sell $(1 \leq j \leq n,\ p_{t_j} \in \mathbb{R}^+ \cup \{0\})$. The tuple $D = (d_1, d_2, \ldots, d_m)$ denotes the amounts of cover money of the bidders where d_i is the amount of cover money of the bidder c_i $(d_i \in \mathbb{R}^+ \cup \{0\})$.

In the DALCOM model, a bid is defined as a pair, $b_k = (R_k, u_k)$, where R_k is the request set and u_k is the upper purchase limit of the bid b_k ($u_k \in \mathbb{Z}^+$). The request set, R_k, consists of z pairs, $R_k = \{(a_{k1}, r_{a_{k1}}), \ldots, (a_{kz}, r_{a_{kz}})\}$, and in each pair $(a_{kl}, r_{a_{kl}})$, a_{kl} denotes the item to be purchased and $r_{a_{kl}}$ denotes the bidder-determined reservation price of the item a_{kl} that indicates the maximum amount for which the owner of the bid b_k is willing to purchase the item a_{kl} ($1 \leq l \leq z$, $a_{kl} \in C$, $r_{a_{kl}} \in \mathbb{R}^+ \cup \{0\}$, $r_{a_{kl}} \geq p_{a_{kl}}$). Finally, the set of bids submitted by the bidder c_i is denoted as B_i, and the set of all bids, $B = \{b_1, b_2, \ldots, b_v\}$, is defined as $B = \bigcup_{i=1}^{m} B_i$.

The meaning of a bid can be stated as follows: By submitting a bid $b_k = (R_k, u_k)$, the bidder c_i declares that he wants to purchase up to u_k items among the items given in $R_k = \{(a_{k1}, r_{a_{k1}}), \ldots, (a_{kz}, r_{a_{kz}})\}$, and for each item a_{kl} the bidder c_i offers to pay at most $r_{a_{kl}}$. If the item a_{kl} is purchased by the bidder c_i, the price of the item a_{kl} is $\mathtt{k} \cdot r_{a_{kl}} + (1 - \mathtt{k}) \cdot p_{a_{kl}}$ where $\mathtt{k} \in [0, 1]$ is the parameter of the k-DA policy. The bid b_k is called *satisfiable* if there exists a nonempty subset of the items in the request set R_k which is available for purchase and the sum of prices of the at most u_k items in this subset is within the limit of the budget of the bidder that is what is spent on purchased items minus the proceeds of the sold items should not exceed the cover money amount of the bidder. The *winner determination problem* (WDP) of the DALCOM model is defined as finding the maximum cardinality set of mutually satisfiable bids such that the total utility, that is the sum of differences of reservation prices of the traded items, is maximized.

In order to formulate the problem using linear integer programming, a binary variable x_{kl} is introduced. It denotes whether the item a_{kl} is purchased in the lth pair of the request set R_k of the bid b_k (1) or not (0). The linear integer programming formulation of the model is as follows:

$$\text{maximize} \sum_{\forall k,l \,|\, b_k \in B \wedge (a_{kl}, r_{a_{kl}}) \in R_k} (r_{a_{kl}} - p_{a_{kl}}) \cdot x_{kl} \tag{5.1}$$

$$\text{subject to} \sum_{\forall k,l \,|\, b_k \in B \wedge (a_{kl}, u_{kl}) \in R_k \wedge a_{kl} = t_j} x_{kl} \leq 1 \qquad (\forall j \,|\, t_j \in T) \tag{5.2}$$

$$\sum_{\forall l \,|\, (a_{kl}, r_{a_{kl}}) \in R_k} x_{kl} \leq u_k \qquad (\forall k |\, b_k \in B) \tag{5.3}$$

$$\sum_{\forall k,l \,|\, b_k \in B_i \wedge (a_{kl}, u_{kl}) \in R_k} (\mathtt{k} \cdot r_{a_{kl}} + (1 - \mathtt{k}) \cdot p_{a_{kl}})\, x_{kl} - \sum_{\forall k,l \,|\, b_k \in B \wedge (a_{kl}, r_{a_{kl}}) \in R_k \wedge a_{kl} \in T_i} (\mathtt{k} \cdot r_{a_{kl}} + (1 - \mathtt{k}) \cdot p_{a_{kl}})\, x_{kl} \leq d_i \qquad (\forall i \,|\, c_i \in C) \tag{5.4}$$

$$x_{kl} \in \{0, 1\} \; (\forall k, l) \tag{5.5}$$

In this formulation, Eq.(5.1) is the objective function which maximizes the total utility. Eq.(5.2) ensures that each item can be purchased by at most one bidder. Eq.(5.3) enforces the upper purchase limits of the bids. Finally, Eq.(5.4) is the budget constraint, that is for each bidder the total cost of the purchased items minus the proceeds of the sold items should not exceed the cover money amount of that bidder.

It should be noted that when the buyer reservation price is equal to the seller reservation price, the utility gain for this transaction will be exactly zero. Thus, this type of transactions may not be selected by the solver software for the optimum solution. In order to prevent this situation, the objective function can be changed as to include a constant such as:

$$\text{maximize} \sum_{\forall k,l \,|\, b_k \in B \wedge (a_{kl}, r_{a_{kl}}) \in R_k} (r_{a_{kl}} - p_{a_{kl}} + \epsilon) \cdot x_{kl}$$

where ϵ is a small positive number which can be defined as:

$$\epsilon = \frac{\min_{k,l} (r_{a_{kl}} - p_{a_{kl}})}{|B| \cdot max_k |R_k|}$$

In this way, we prevent the disturbance caused by this constant to affect the optimum solution.

The linear integer program for the example scenario introduced in Section 5.2 and the corresponding solution can be seen in Figure 5.2.

5.3.1. Theorems Related to the Winner Determination Problem

In this section, we present some complexity results for the WDP.

Proposition 5.1. *The decision version of the WDP is NP-complete.*

Proof. Let Π be the decision version of the WDP. Π is defined as follows: Given a set of bidders C, a set of items T and associated reservation prices P, a set of cover money amounts D, a set of bids B and a positive integer K, is there a subset $B' \subseteq B$ such that the following inequalities are satisfied?

$$\sum_{\forall k,l \,|\, b_k \in B' \wedge (a_{kl}, r_{a_{kl}}) \in R_k} (r_{a_{kl}} - p_{a_{kl}}) \geq K \tag{5.6}$$

$$\sum_{\forall k,l \,|\, b_k \in B' \wedge (a_{kl}, u_{kl}) \in R_k \wedge a_{kl} = t_j} 1 \leq 1 \quad (\forall j \,|\, t_j \in T) \tag{5.7}$$

$$\sum_{\forall l \,|\, (a_{kl}, r_{a_{kl}}) \in R_k} 1 \leq u_k \quad (\forall k |\, b_k \in B') \tag{5.8}$$

$$\sum_{\forall k,l \,|\, b_k \in B_i \cap B' \wedge (a_{kl}, u_{kl}) \in R_k} (\mathtt{k} \cdot r_{a_{kl}} + (1-\mathtt{k}) \cdot p_{a_{kl}}) - \sum_{\forall k,l \,|\, b_k \in B' \wedge (a_{kl}, r_{a_{kl}}) \in R_k \wedge a_{kl} \in T_i} (\mathtt{k} \cdot r_{a_{kl}} + (1-\mathtt{k}) \cdot p_{a_{kl}}) \leq d_i \quad (\forall i \,|\, c_i \in C) \tag{5.9}$$

If we have a certificate that consists of set B', this certificate can be verified in polynomial time by checking Eq.(5.6-5.9). Therefore, Π is in NP.

```
Maximize
10 x11 + 5 x12 + 10 x21 + 5 x31 + 5 x32 + 10 x33 + 0 x41 + 5 x42
 + 10 x43 + 5 x51

Subject To
Book A : x31 + x41 <= 1
Book B : x32 <= 1
Book C : x11 + x42 <= 1
Book D : x21 <= 1
Book E : x12 + x33 + x43 + x51 <= 1

Bid 1 : x11 + x12 <= 1
Bid 2 : x21 <= 1
Bid 3 : x31 + x32 + x33 <= 2
Bid 4 : x41 + x42 + x43 <= 1
Bid 5 : x51 <= 1

Bidder 1 : 30 x11 + 32.5 x12 + 35 x21 - 42.5 x31 - 40 x41
            - 32.5 x32 <= 0
Bidder 2 : 42.5 x31 + 32.5 x32 + 35 x33 - 30 x11 - 27.5 x42 <= 45
Bidder 3 : 40 x41 + 27.5 x42 + 35 x43 - 35 x21 <= 0
Bidder 4 : - 32.5 x12 - 35 x33 - 35 x43 - 32.5 x51 <= 0
Bidder 5 : 32.5 x51 <= 35

Solution
Objective Value = 40
x11 = 1    x12 = 0    x21 = 1    x31 = 1    x32 = 1
x33 = 0    x41 = 0    x42 = 0    x43 = 1    x51 = 0
```

Figure 5.2. Linear integer program for the example market scenario in Figure 5.1 and its solution. All variables are binary variables.

Next, we present a polynomial time transformation from the subset sum problem. Let Π' be the subset sum problem [50, p. 243] which is defined as follows: Given a finite set of positive integers $A = \{a_1, a_2, \ldots, a_f\}$ and a positive integer B, is there a subset $A' \subseteq A$ such that $\sum_{\forall i\,|a_i\,\in A'} a_i = B$?

Let $\Pi'(A, B)$ be an instance of the subset sum problem. It can be transformed to Π in polynomial time as follows: Let the set of bidders C consist of two bidders ($C = \{c_1, c_2\}$), and the set of items to be sold by the bidder c_1 be $T_1 = \{t_1, t_2, \ldots, t_f\}$ and by the bidder c_2 be $T_2 = \{t_{f+1}\}$ ($f = |A|$). The reservation prices of all the items declared by their owners are zero ($P = (0, \ldots, 0)$). Let the cover money amounts of all the bidders be zero ($D = \{0, 0\}$). Assume that the bidder c_1 submits only one bid requesting one item $b_1 = \langle\{(t_{f+1}, B/k)\}, 1\rangle$ and the bidder c_2 also submits one bid requesting up to f items $b_2 = \langle\{(t_1, a_1/k), (t_2, a_2/k), \ldots, (t_f, a_f/k)\}, f\rangle$. Finally, let $K = 2kB$.

Since the cover money amounts of the bidders are zero, the bids b_1 and b_2 cannot be satisfied alone. Together, they can only be satisfied if and only if there is a solution to the subset sum problem instance $\Pi'(A, B)$. Therefore, the solution of the problem instance of Π is also a solution of the problem instance of Π' and vice versa.

Since Π is in NP and the subset sum problem is NP-complete, the decision version of the WDP is NP-complete. □

Using this proof, we can also show the inapproximability of the WDP.

Lemma 5.2. *Unless P=NP, there is no polynomial time ϵ-approximative algorithm for the WDP for any $\epsilon \in [0, 1)$.*

Proof. If there exists a polynomial time ϵ-approximative algorithm for the WDP for any $\epsilon \in [0, 1)$, this algorithm can be used for deciding the subset sum problem as described in the proof of Proposition 5.1. Since the subset sum problem is NP-Complete, this would imply P=NP. □

This inapproximability result suggests that finding a non-trivial feasible solution of a WDP instance or proving that no such solution exists can be difficult. However, this result is valid for a special subset of the WDP instances in which the amounts of cover money are zero for all bidders. It is obvious that if at least one bidder has enough cover money to purchase at least one of the items in one of his bids, then finding a nonzero feasible solution becomes a polynomial-time problem. Also, if the cover money amounts of all bidders allow them to purchase every item they bid for without selling their items, in other words the budget constraint in Eq. 5.4 is relaxed in the WDP, and then this new problem can be solved in polynomial-time. We call this problem as *relaxed-budget WDP problem* (RBP). A fast polynomial-time algorithm for solving RBP is proposed below.

5.3.2. An Algorithm for Solving the Relaxed-Budget WDP Problem

We model RBP as a minimum cost flow problem [106]. Let $N(V, A, l, u, c, b)$ denote a network with node set V, arc set A, lower bound $l(v, w)$, upper bound $u(v, w)$ and cost $c(v, w)$ for each arc $(v, w) \in A$, and supply/demand values $b(v)$ for each node $v \in V$.

We begin constructing the network by introducing a node for each bid $b_k \in B$ and for each item $t_j \in T$. The source node of the network is represented with s and the sink node is represented with t. For each bid $b_k \in B$, an arc is drawn from the the source node s, to the bid node b_k. The upper bound of this arc is the upper purchase limit of the corresponding bid, that is $u(s, b_k) = u_k$, and the associated cost is zero ($c(v, w) = 0$). Then, for each bid $b_k \in B$ and for each pair $(a_{kl}, r_{a_{kl}}) \in R_k$, an arc is drawn from the node b_k to a_{kl}. The upper bound of this arc is one, that is $u(b_k, a_{kl}) = 1$, and the cost of the arc is the additive inverse of the utility value ($c(b_k, a_{kl}) = p_{a_{kl}} - r_{a_{kl}}$). Finally, for each item $t_j \in R$, we introduce an arc from the node t_j to the sink node t with upper bound of one and cost of zero ($u(t_j, t) = 1$, $c(t_j, t) = 0$). There is no flow requirement for any arc ($l(v, w) = 0 \; \forall v, w | (v, w) \in A$) and there is no supply or demand for any node in the network, therefore, $b(v) = 0$ for every node $v \in V$.

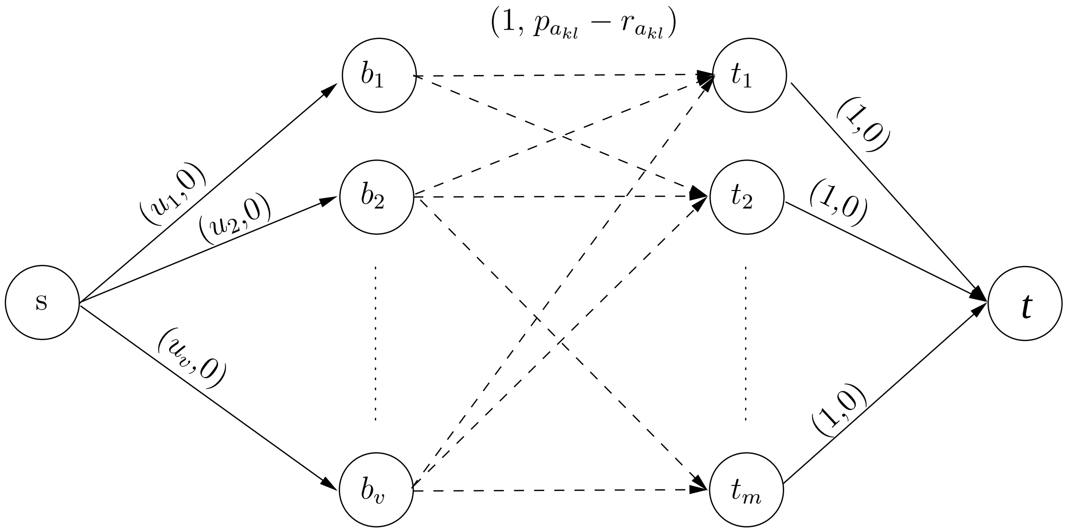


Figure 5.3. The network of the RBP with (capacity,cost) values on the arcs.

The constructed network is shown in Figure 5.3. The minimum cost flow in this network gives the optimum solution for RCP since the constraints in Eq.(5.2,5.3) are encoded in the network and minimizing the cost in the network means maximizing the total utility.

5.4. Test Case Generator

As proven in Section 5.3.1, when all the bidders have no cover money at all, the problem instances are difficult to solve. In fact, most of the problem instances of this kind would have no nonzero feasible solutions at all. The reason is that in all nonzero feasible solutions of such an instance, for *every* bidder the proceed of the sold items should be equal to the amount spent on the purchased items which is very unlikely to occur in the practice. However, this situation can easily be prevented by the market maker who operates the market by forcing a lower limit for the cover money amounts of bidders if it is necessary. On the other hand, when all the bidders have enough money to cover all of their purchases, in this case finding the optimum solutions for these problem instances becomes easier requiring only polynomial time. Actually, the problem instances that arise from the real markets would mostly be in between these two endpoints. So, the question is "How difficult would it be to solve the real market problem instances?". In order to estimate this, we have designed a test case generator and prepared a test suite. In this section, we explain the test case generator in detail, and in the following section, the test suite and the experimental results are presented.

The test case generator uses GNU Scientific Library [138] and supports all common continuous and discrete random number distributions. It has ten parameters which can be configured to design a factorial testing. These parameters are as follows:

(i) *Market Type:* This parameter defines the market type to which the problem instances belong. The market type determines which types of items are available in the market and their price ranges. The generator supports three different market types, namely *Book*, *CD/DVD* and *Electronic* markets. The available item types in these markets and their price profiles can be seen in Table 5.1. These statistical price profiles for book, CD/DVD and electronic markets are obtained from the studies of Ghose et al. [135], Smith et al. [134] and Ghose [139], respectively. They are based on the sales information in the Amazon.com marketplace.

(ii) *Number of Bidders:* This parameter defines the number of bidders in the market. It accepts a non-empty set of constant numbers.

(iii) *Number of Items per Bidder:* This parameter defines the number of items that each bidder put up for sale. It accepts a non-empty set of discrete distributions.

(iv) *Number of Bids per Bidder:* This parameter defines the number of purchase bids placed by each bidder. It accepts a non-empty set of discrete distributions.

(v) *Request Set Determination Method:* There two methods for determining the requested items in a bid. In the *uniform selection method*, the requested items in a bid are selected uniformly among the available items in the market. In the *closely priced items selection method*, on the other hand, only the first requested item is selected uniformly among the available items and the rest are selected uniformly from a set of items whose value are close to the value of the first item. The motivation behind this second method is that the values of substitutable items are generally close. For instance, it is very unlikely for a copy of a physics book to be sold for one dollar and another copy for a hundred dollars.

(vi) *Request Set Size:* This parameter defines the number of items requested in a bid. It accepts a non-empty set of discrete distributions.

(vii) *Upper Purchase Limit:* This parameter defines the upper purchase limit of each bid. It accepts a non-empty set of discrete distributions.

(viii) *Cover Money Determination Method:* There two methods for determining cover money amounts. In the *bids first method*, after the items a bidder wants to sell are determined, purchase bids of that bidder are generated before determining his cover money amount. Then, based on the prices of the items he wants to sell and the amount required for his purchase bids, the actual cover money amount is determined. This method is used for representing the bidders who first determine which items to buy and then the cover money amount.

However, in the *cover money first method*, after the items a bidder wants to sell are determined, his cover money amount is determined before generating purchase bids. The purchase bids are then generated based on the maximum budget of the bidder which is the sum of the reservation prices of the items he want to sell and his cover money amount. This method is used for representing the bidders who select the items to be bought based on their budget.

(ix) *Cover Money Ratio:* This parameter defines the ratio that is used for determining the amounts of cover money of the bidders. It accepts a non-empty set of continuous distributions. The amount of cover money d_i for a bidder c_i is calculated as:

$$d_i = cmr_i \cdot (cm_i^{max} - cm_i^{min}) + cm_i^{min}$$

where cmr_i is the cover money ratio, cm_i^{min} and cm_i^{max} are the minimum and the maximum amount of cover money required for the bidder c_i, respectively. Based on the selected cover money determination method, cm_i^{min} and cm_i^{max} values may differ.

If the bids first method is used, since the purchase bids are generated before determining the cover money amount, cm_i^{min} is defined as the difference between the highest reservation price offered by the bidder for an item and the sum of the reservation prices of the items he sells. This lower bound ensures that the bidder can afford any *one* of requested items in his bids in case all of his items are sold. cm_i^{max} is defined as the maximum amount of money required for the bidder if all of his bids are satisfied and none of his items are sold.

If the cover money first method is used, since the cover money amount is determined first, no positive lower bound is required for cm_i^{min} and its value is simply zero. However, since the bids are generated afterwards, cm_i^{max} is defined as

$$cm_i^{max} = Avg\_\#\_Bids\_per\_Bidder \cdot Avg\_Req\_Set\_Size \cdot Avg\_Item\_Price$$

where $Avg\_\#\_Bids\_per\_Bidder$ and $Avg\_Req\_Set\_Size$ are the mean values of the distributions supplied to the parameters *Number_of_Bids_per_Bidder* and *Request_Set_Size*, respectively. Avg_Item_Price is the average price of the items sold in the market calculated using the price profile which corresponds to the selected market type. Thus, if a bidder has a cover money of amount cm_i^{max}, he would be able place $Avg\_\#\_Bids\_per\_Bidder$ bids with $Avg\_Req\_Set\_Size$ requested items per bid on average in accordance with the distributions supplied to the respective parameters. Furthermore, he may be able to purchase the requested items in his bids even none of his items are sold as in the case of the bids first method.

In summary, independent of the used cover money determination method, when the cover money ratio is close to zero, the amount of cover money of a bidder also tends to zero. On the contrary, when the cover money ratio is close to one, the amount of cover money of a bidder tends to a value which allows him to purchase the items he wants without the need of proceeds of the his sold items.

(x) *k:* This is the parameter of k-DA policy. It accepts a real number in the range $[0, 1]$.

The algorithm of the test case generator is straightforward. First, for each bidder the items to be sold and the respective seller reservation prices are determined. The items are selected uniformly among the item types available in the market. In order to determine the seller reservation price p_i for an item t_i, a minimum price p_i^{min} and a

Table 5.1. Available item types and their price profiles for the test case generator. All the price values are in US$.

		Lower Bound of the Price				*Upper Bound of the Price*			
Market Type	*Item Type*	*mean*	*stdev*	*min*	*max*	*mean*	*stdev*	*min*	*max*
Book	Like New	15.16	21.32	0.01	194.25	24.04	26.47	1.95	209.99
	Very Good	11.26	18.41	0.01	207.60	24.04	26.47	1.95	209.99
	Good	11.24	16.42	0.01	200.00	24.04	26.47	1.95	209.99
	Acceptable	7.86	15.70	0.01	222.35	24.04	26.47	1.95	209.99
CD/DVD	CD	9.10	9.28	0.55	99.99	16.00	11.28	2.98	119.49
	DVD	17.16	19.45	0.85	149.99	28.57	22.97	7.98	159.99
Electronic	PDA	262.56	161.32	0.99	1,049.99	599.59	245.03	29.61	2,298.99
	Digital Camera	415.14	328.89	0.88	7,999.99	1,351.52	1,068.84	82.78	7,999.99
	Audio Player	162.93	126.96	1.00	499.95	467.61	207.45	35.02	499.95
	Laptop	988.87	397.89	9.24	1,999.99	1,486.73	617.96	74.88	1,999.99

maximum price p_i^{max} are drawn based on the statistical price profile given in Table 5.1. Then, the seller reservation price p_i is calculated as

$$p_i = \mathcal{N}(\bar{p_i}, (p_i^{max} - \bar{p_i})/2)$$

where $\bar{p_i} = (p_i^{max} + p_i^{min})/2$ and $\mathcal{N}$ is the normal distribution. After the items and their reservation prices are determined, depending on the cover money determination method, the amounts of cover money are determined first, and then the purchase bids are generated or vice versa. The buyer reservation prices for the requested items are also determined as the seller reservation prices with one difference and that is the buyer reservation prices should be greater than equal to the seller reservation prices since negative utility values are not allowed. This also ensures that all the bids and the pairs in the bids are valid and no preprocessing is necessary. Thus, the number of generated bids and the size of the request sets in the generated instances are consistent with the *Number_of_Bids_per_Bidder* and *Request_Set_Size* parameters which is important for correct evaluation of the experimental results. For further details on the test case generator, the source code is available from [140].

Table 5.2. The values of the parameters used in the test case generator.

Parameter Name	*Values*
Market Type	Book, CD/DVD, Electronic
Number of Bidders	100, 250, 500, 1000, 2000, 10000 25000
Number Of Items per Bidder	Poisson (2), Poisson (4), Poisson (6)
Number of Bids per Bidder	Poisson (2), Poisson (4), Poisson (6)
Request Set Determination Method	Closely Priced, Uniform Selection
Request Set Size	Poisson (2), Poisson (4), Poisson (6)
Upper Purchase Limit	Discrete Uniform (1, Req. Set Size), Poisson (1.5)
Cover Money Determination Method	Cover Money First, Bids First
Cover Money Ratio	Normal (0.05, 0.05), Normal (0.1, 0.1), Normal (0.25, 0.25), Normal (0.5, 0.5), Normal (0.75, 0.75)
k-DA Parameter	0.5

5.5. Experimental Results

In order to estimate the performance of a general purpose MIP solver for the WDP on real-world markets, we have prepared a comprehensive test suite consisting of 22,680 WDP instances. The values of the parameters used for generating test suite can be seen in Table 5.2. The test suite supports full-factorial testing such that all possible combinations of the values of the parameters can be tested.

In this test suite, the number of bidders is varied between 100 and 25,000 in order to represent small to very large markets. Each bidder is assumed to put forward a number of items for sale as well as place a number of bids for purchase both of which are assumed to be distributed with Poisson distribution having mean values of two, four and six. Therefore, instances with different ratios of the number of sale requests to purchase requests are generated. The same distributions are also used for the *Request_Set_Size* parameter in order to simulate different levels of bidders' indifference to the items they want to purchase. In accordance with the sizes of their request sets, the upper purchase limits of the bids are assumed to be distributed with either Poisson distribution having a mean value of 1.5 or discrete uniform distribution having a lower bound of one and an upper bound as the size of the request set. The former distribution is used to represent the case in which the bidders are willing to buy a few items at most, and the latter is used to represent the more general cases in which the bidders are interested in half of the items listed in their request set on average. Finally, five different average cover money ratios between 5% - 75% are used to observe the effect of different cover money amounts.

The generated test cases were solved using the MIP solver in the IBM ILOG CPLEX Optimizer package version 12.2 [55]. The solver was configured to use eight threads in parallel and each instance was run on eight hyper-threaded CPU cores (on four physical CPU cores) of 2.93 GHz clock speed with 16 GB of memory. Also, a wall-clock time limit of 30 minutes was defined for each instance. The other parameters of the solver were left untouched. The operating system used was 64 bit Linux.

Among the 22,680 instances, the CPLEX MIP solver found the optimum solution for 17,278 instances. For 4,697 instances, the solver could not find the optimum solution within the time limit, however, it was able to find a nonzero feasible solution. For the rest of the 705 instances, the solver was unable to find a solution within the time limit. The instances for which the solver could not find a solution within the time limit are referred as insolvable instances.

The results classified based on the parameters *Number_of_Bidders*, *Cover_Money_Determination_Method* and *Cover_Money_Ratio* can be seen in Table 5.3. The other parameters are left free.

Table 5.3. The number of optimum solutions (opt), feasible solutions (feas) and insolvable instances (insol) classified based on the parameters *Number_of_Bidders*, *Cover_Money_Determination_Method* and *Cover_Money_Ratio*.

	Number of Bidders	100		250		500		1000		2000		10000			25000		
CM Det. Met.	*Avg. CM Ratio*	*opt*	*feas*	*opt*	*feas*	*opt*	feas	opt	feas	opt	feas	opt	*feas*	*insol*	*opt*	*feas*	*insol*
CM First	5%	255	69	169	155	130	194	102	222	82	242	52	204	68	41	185	98
	10%	323	1	244	80	191	133	155	169	138	186	109	156	59	83	147	94
	25%	324	0	322	2	312	12	282	42	243	81	194	116	14	163	91	70
	50%	324	0	324	0	324	0	322	2	314	10	279	45	0	236	73	15
	75%	324	0	324	0	324	0	324	0	323	1	307	17	0	290	33	1
	Overall	**1550**	**70**	**1383**	**237**	**1281**	**339**	**1185**	**435**	**1100**	**520**	**941**	**538**	**141**	**813**	**529**	**278**
Bids First	5%	271	53	211	113	186	138	156	168	136	188	117	161	46	110	141	73
	10%	311	13	267	57	227	97	190	134	182	142	154	129	41	136	115	73
	25%	324	0	324	0	307	17	292	32	272	52	244	69	11	203	93	28
	50%	324	0	324	0	324	0	323	1	314	10	301	23	0	271	41	12
	75%	324	0	324	0	324	0	324	0	324	0	310	14	0	294	28	2
	Overall	**1554**	**66**	**1450**	**170**	**1368**	**252**	**1285**	**335**	**1228**	**392**	**1126**	**396**	**98**	**1014**	**418**	**188**
Overall		**3104**	**136**	**2833**	**407**	**2649**	**591**	**2470**	**770**	**2328**	**912**	**2067**	**934**	**239**	**1827**	**947**	**466**

It is observed that for the instances containing up to 2,000 bidders, the performance of the solver is quite good, although some of the solutions are not optimum. However, among the instances with 10,000 and 25,000 bidders, 705 of them are insolvable. The number of optimum solutions is also smaller for these instances. Another observation is related to the cover money ratio. As expected, the difficulty of the instances increases dramatically as the cover money ratio decreases. When the average cover money ratio is 75%, almost all the instances are solvable optimally within the time limit. However, when the ratio is 5%, the solver could not find the optimum solution for 122 instances out of 624 instances containing only 100 bidders. Finally, the cover money determination method is observed as being less effective than the other two parameters. The instances generated using the bids first approach are slightly easier than the instances generated using the cover money first approach. The reason is that the minimum amounts of cover money for some bidders are strictly positive in the bids first approach rather than being zero as in the cover money first approach as explained in Section 5.4. The running times of the solver for the optimum solutions are also presented in Table 5.4. The results in this table also follow almost the same pattern. The slight differences are caused by not including the running times of the solver for both the feasible and the insolvable instances.

Table 5.5 presents the results which are classified based on the parameters *Number_of_Items_per_Bidder*, *Number_of_Bids_per_Bidder* and *Request_Set_Size*. It is observed that as the average number of items per bidder increases, the instances become easier to solve. For instance, when the average number of items per bidder is 2, the solver finds the optimum solution for the 5,247 instances out of 7,560 instances and there are 546 insolvable instances. However, when the average number of items per bidder is 6, optimum solutions for 6,316 instances are found and there are only 12 insolvable instances. In contrast to the average number of items per bidder, the difficulty of the problem instances increases as the average number of bids per bidder increases. Therefore, the ratio of the average number of items per bidder to the average number of bids per bidder has a particular effect on the difficulty of the WDP. The effect of the average request set size is, on the other hand, not conclusive on its own, and it is based on this ratio. When this ratio is small, the instances become more difficult to solve as the average request set size increases whereas when the ratio is high, this has no negative effect on the difficulty of the instances, and even slight improvements are observed.

The results which are classified based on the rest of the parameters, i.e. the *Market_Type*, *Request_Set_Determination_Method* and *Upper_Purchase_Limit*, can be seen in Table 5.6. It is observed that the numbers of optimum, feasible and insolvable instances are almost same across the different configurations, and therefore it can be concluded that the values of these parameters have no significant effect on the difficulty of the instances for the CPLEX MIP solver.

Finally, Table 5.7 presents the histogram of the goodness of the feasible solutions found by the CPLEX MIP solver classified based on the number of bidders. The *goodness* of a feasible solution is defined as

$$Goodness \;=\; \frac{The\ objective\ value\ of\ the\ feasible\ solution}{The\ best\ upper\ bound\ of\ the\ objective\ function}$$

For the instances for which the solver could not find an upper bound, the objective value of the linear relaxation of the corresponding instance was used as the upper bound. It is seen that for almost half of the feasible solutions, the objective values are very close to the optimum, i.e. within at least 99% of the objective value of the corresponding optimum solution. If we consider a goodness of 0.9 as a threshold, then 2,843 out of 4,697 instances become solvable corresponding to approximately 60% of the feasible solutions. This ratio increases to approximately 96% as the threshold decreases to 0.5 and only there are 200 instances which are below this threshold. The plot of the more detailed histogram can be seen in Figure 5.4.

Table 5.4. The running times (s) of the CPLEX MIP solver for the optimum solutions classified based the parameters *Number_of_Bidders*, *Cover_Money_Determination_Method* and *Cover_Money_Ratio*.

	Number of Bidders	100		250		500		1000		2000		10000		25000	
CM Det. Met.	*Avg. CM Ratio*	*mean*	*stdev*	*mean*	*stdev*	*mean*	*stdev*	*mean*	*stdev*	*mean*	*stdev*	*mean*	*stdev*	*mean*	*stdev*
CM First	5%	71	229	60	209	51	175	76	222	144	315	109	111	352	221
	10%	32	161	51	195	53	210	61	213	58	167	130	260	211	269
	25%	0	1	5	21	34	149	49	168	48	131	81	138	218	268
	50%	0	0	1	1	2	5	10	58	25	103	66	134	191	292
	75%	0	0	0	0	1	1	3	4	12	48	44	80	178	297
	Overall	**19**	**121**	**18**	**113**	**22**	**124**	**30**	**136**	**39**	**139**	**72**	**143**	**202**	**286**
Bids First	5%	35	135	41	162	51	182	57	193	70	224	50	160	100	143
	10%	27	154	45	154	67	228	33	152	91	272	78	185	192	335
	25%	1	3	17	101	20	138	26	129	34	109	80	166	142	174
	50%	0	0	1	13	1	3	9	66	12	55	60	98	168	268
	75%	0	0	0	0	1	1	2	3	10	27	41	53	152	202
	Overall	**12**	**90**	**18**	**104**	**23**	**134**	**20**	**114**	**35**	**145**	**61**	**130**	**154**	**234**
Overall		**15**	**107**	**18**	**108**	**23**	**129**	**25**	**125**	**37**	**142**	**66**	**136**	**175**	**259**

Table 5.5. The number of optimum solutions (opt), feasible solutions (feas) and insolvable instances (insol) classified based on the parameters *Number_of_Items_per_Bidder*, *Number_of_Bids_per_Bidder* and *Request_Set_Size*.

Avg. # Items Per Bidder	*Avg. # Bids Per Bidder*	*Avg. Req. Set Size*	*opt*	*feas*	*insol*
2	2	2	794	44	2
		4	776	62	2
		6	795	44	1
		Overall	**2365**	**150**	**5**
	4	2	646	142	52
		4	519	240	81
		6	461	293	86
		Overall	**1626**	**675**	**219**
	6	2	576	187	77
		4	393	330	117
		6	287	425	128
		Overall	**1256**	**942**	**322**
	Overall		**5247**	**1767**	**546**
4	2	2	805	33	2
		4	828	12	0
		6	839	1	0
		Overall	**2472**	**46**	**2**
	4	2	635	192	13
		4	627	208	5
		6	664	176	0
		Overall	**1926**	**576**	**18**
	6	2	508	275	57
		4	417	373	50
		6	392	428	20
		Overall	**1317**	**1076**	**127**
	Overall		**5715**	**1698**	**147**
6	2	2	826	14	0
		4	840	0	0
		6	840	0	0
		Overall	**2506**	**14**	**0**
	4	2	695	144	1
		4	741	99	0
		6	777	63	0
		Overall	**2213**	**306**	**1**
	6	2	528	304	8
		4	520	318	2
		6	549	290	1
		Overall	**1597**	**912**	**11**
	Overall		**6316**	**1232**	**12**

Table 5.6. The number of optimum solutions (opt), feasible solutions (feas) and insolvable instances (insol) classified based on the parameters *Market_Type*, *Request_Set_Determination_Method* and *Upper_ Purchase_Limit*.

Market Type	*Req. Set Det. Method*	*Upper Pur. Limit*	*Number of Solutions* opt	feas	insol
Book	Closely Priced	Uniform	1367	474	49
		Poisson	1398	437	55
		Overall	**2765**	**911**	**104**
	Uniform Selection	Uniform	1409	414	67
		Poisson	1425	385	80
		Overall	**2834**	**799**	**147**
	Overall		**5599**	**1710**	**251**
CD/DVD	Closely Priced	Uniform	1464	388	38
		Poisson	1437	401	52
		Overall	**2901**	**789**	**90**
	Uniform Selection	Uniform	1481	343	66
		Poisson	1465	358	67
		Overall	**2946**	**701**	**133**
	Overall		**5847**	**1490**	**223**
Electronic	Closely Priced	Uniform	1437	408	45
		Poisson	1448	388	54
		Overall	**2885**	**796**	**99**
	Uniform Selection	Uniform	1471	360	59
		Poisson	1476	341	73
		Overall	**2947**	**701**	**132**
	Overall		**5832**	**1497**	**231**

Table 5.7. Number of feasible solutions classified according to the goodness values.

	Goodness			
Number of Bidders	0.00 - 0.50	0.50 - 0.90	0.90 - 0.99	0.99 - 1.00
100	0	0	29	107
250	0	0	95	312
500	0	18	143	430
1000	15	116	131	508
2000	96	337	47	432
10000	58	584	13	279
25000	31	599	39	278
Overall	**200**	**1654**	**497**	**2346**

5.6. Discussion and Conclusion

In this chapter, we have proposed an e-market model designed especially for secondary markets for the trading of used goods and as well as new ones. The model is based on the discrete-time DA institution in which each bidder can both put forward items for sale and place bids for purchase during the predefined submission period. The model augments the DA institution with a cover money mechanism which allows bidders to declare an amount of cover money in order to limit the maximum amount of money to be spent in the market. Thus, it is ensured that for each bidder what is spent on purchased items minus the proceeds of sold items do not exceed the declared cover money amount. Furthermore, the model further provides a mechanism in which the bidders are allowed to declare a list of substitutable items and put a limit on the number of items to be purchased in that list. These mechanisms provide a higher trading volume and a better allocation of items by encouraging bidders to participate in the e-market without the risk of having a budget deficit and by enabling bidders to express their preferences of substitutabilities in the market.

We have formally defined the winner determination problem using linear integer programming and proven that the decision version of this problem is NP-complete. Furthermore, it is shown that the problem is inapproximable unless $P = NP$. However, this theoretical result is valid if the cover money amounts of all the bidders are zero which is very unlikely to occur in a real market and also which can easily be prevented by the market maker. Therefore, in order to estimate the real-life performance of the model, we have designed a test case generator and prepared a comprehensive test suite. The test suite is then solved using the CPLEX MIP solver. Within a time limit of 30 minutes, the solver has found optimum solutions for approximately 76% of the instances. For the instances in which the number of bidders is not greater than 2,000, this ratio increases to 82%. Besides, if a goodness value of 0.5 is considered as adequate, then the CPLEX MIP solver is capable of solving 96% of these instances. Thus, this general purpose MIP solver can readily be used for solving the winner determination problem in small markets, although there is room for improvement. For larger markets consisting of more than 10,000 bidders, on the other hand, the solver could not find any solution at all for approximately 11% of the instances. Besides, the average goodness of the feasible solutions are also lower compared to the smaller problem instances.

As demonstrated by the experiments, the difficulty of the problem instances tends to decrease as the cover money amounts increase and the ratio of the number of items to be sold per bidder to the number of bids per bidder increases. Thus, the market maker may enforce or motivate bidders to declare higher amounts of cover money and put forward higher number of items for sale and/or place less number of bids. Although the difficult problem instances can be avoided to a degree by these restrictions, they would reduce the allocative efficiency of the market. Therefore, as a future work, designing heuristic methods which would be used as a complement to the MIP solver for solving unrestricted difficult problem instances should be considered.

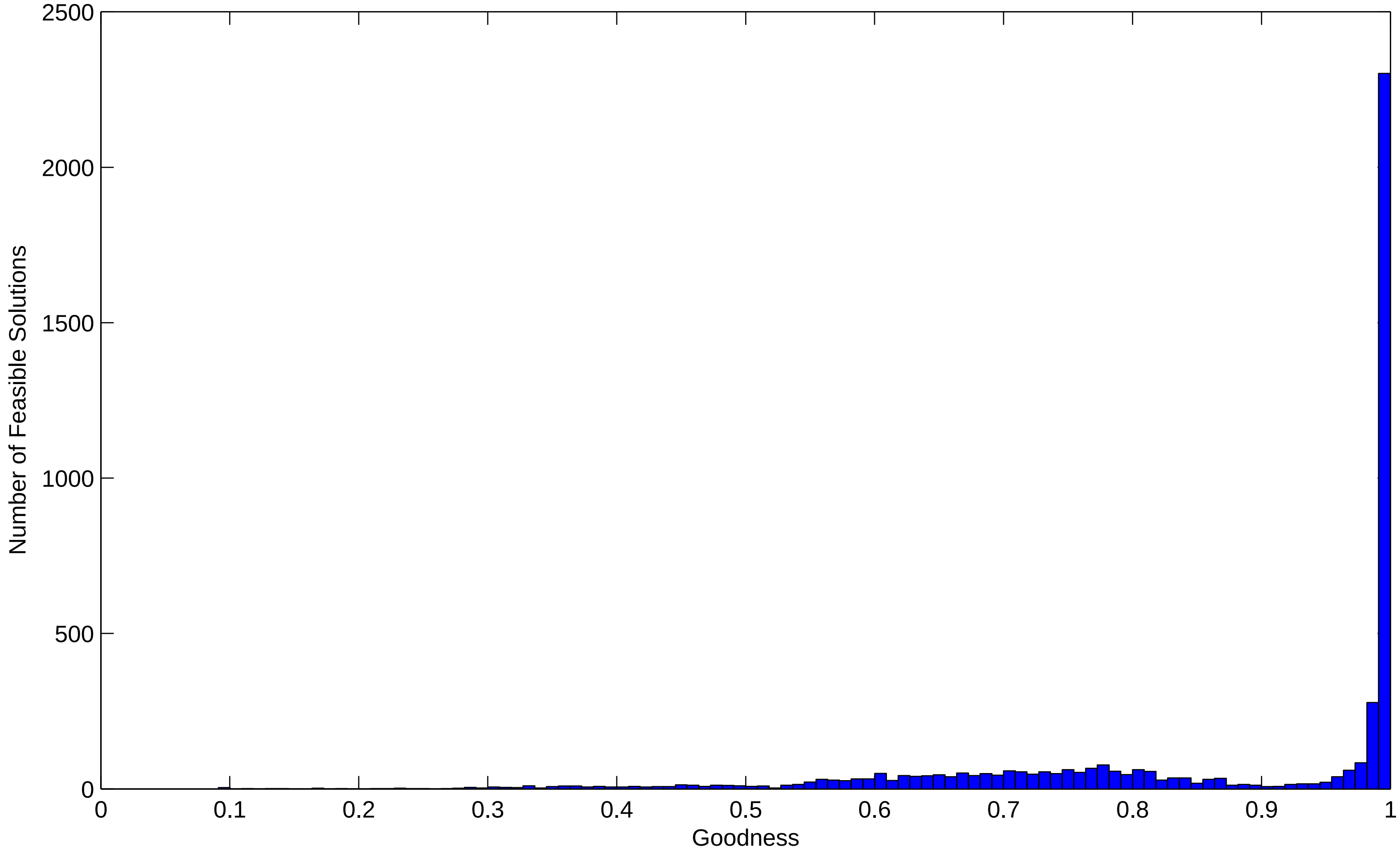


Figure 5.4. The histogram plot of the goodness of the feasible solutions.

6. CONCLUSION

Recent advances in information and communication technologies provided us a platform so that we now have:

- high computational power making it possible to solve large scale problems pretty quickly,
- the Internet providing a worldwide communication medium for the people, computers and mobile devices, and
- the World Wide Web providing a high level user interface for the users worldwide.

This platform provides most of the components needed for building large-scale e-markets in which participants worldwide can use in order to engage in trade transactions. Other important components that are needed are market models and algorithmic engines that analyze the bids of the participants and extract from the collection of bids, feasible trading patterns that optimize some criteria such as the trading volume and/or the social welfare.

In this thesis, we contribute three auction and barter based e-market models and corresponding optimization algorithms that provide better allocation of items and higher social welfare. Our first model, i.e. the direct barter model for the course add/drop process, utilizes direct bartering institution without using money as a medium of exchange for improving the efficiency of the course add/drop process in the universities. In addition to the usual add and drop requests, the model allows students to barter their courses for the courses they want. This feature prevents possible inefficiencies and dead-locks which are seen in the first come first served based systems. In our model, we also introduce a two-level weighting system that enables students to express priorities among their requests while providing fairness among the students. The benefits of introducing these mechanisms and the fast performance of our optimization algorithm have been demonstrated on various test cases based on the registration data of our university.

In our second model, i.e. the multi-unit differential auction barter model, we extend the direct bartering institution by introducing differential payment mechanism. This mechanism allows bidders to barter different valued items by declaring a differential payment amount which is the amount of money the bidder is willing to get or give if the bartering takes place. Besides, the usual sale and purchase bids in a DA institution are also preserved. The model further comprises a powerful and flexible bidding language. This language allows bidders to express their complex preferences of purchase, sell and exchange requests for multiple instances of commodities, and hence increases the allocative efficiency of the market. A fast algorithm is proposed for the optimization problem of the model and its performance is demonstrated on various test cases containing up to one million bids. Thus, the model can be used even in large-scale online auctions without worrying about the running times of the solver.

Finally, in our third model, i.e. the double auction with limited cover money model, we extend the discrete time double auction institution with a mechanism which allows bidders to declare an amount of cover money for limiting their transactions. Thus, for each bidder what is spent on purchased items minus the proceeds of sold items cannot exceed his cover money amount. This feature encourages bidders to participate in the e-markets without the risk of having a budget deficit. Besides, since a bidder may also be indifferent to multiple items, the model further provides a mechanism for expressing their preferences of substitutabilities in the market which results in a better allocation of resources. However, different from our first and second models, the decision version of the optimization problem for this model is NP-complete. Therefore, we have also designed a test case generator with wide-range of configurable parameters and prepared a comprehensive test suite. We have demonstrated the performance of the CPLEX MIP solver on this test suite. We have observed that although the complexity results are discouraging, the performance of the solver for the cases containing up to 2000 bidders is satisfactory making the model suitable to be used in e-markets of that size. However, designing fast heuristic methods would be necessary for operating larger markets.

REFERENCES

1. McMillan, J., *Game Theory*, W. W. Norton & Company, USA, 2003.
2. Bakos, Y., "The Emerging Role of Electronic Marketplaces on the Internet", *Commun. ACM*, Vol. 41, pp. 35–42, 1998.
3. van Heck, E. and P. M. A. Ribbers, "Experiences with Electronic Auctions in the Dutch Flower Industry", *Electronic Markets*, Vol. 7, No. 4, 1997.
4. Bakos, J. Y., "A Strategic Analysis of Electronic Marketplaces", *MIS Q.*, Vol. 15, pp. 295–310, 1991.
5. Grieger, M., "Electronic Marketplaces: A Literature Review and a Call for Supply Chain Management Research", *European Journal of Operational Research*, Vol. 144, No. 2, pp. 280 – 294, 2003.
6. *Who We Are - eBay Inc.*, `http://www.ebayinc.com/who`, May 2011.
7. *Alibaba.com Q4 2010 and FY 2010 Results*, `http://img.alibaba.com/images/cms/upload/alibaba_group/about_alibaba/b2b_releases/20110317_alibaba_q4_and_fy_2010_results_eng.pdf`, May 2011.
8. Kambil, A., P. F. Nunes and D. Wilson, "Transforming the Marketspace with All-In-One Markets", *Int. J. Electron. Commerce*, Vol. 3, pp. 11–28, 1999.
9. Bakos, J. Y., "Reducing Buyer Search Costs: Implications for Electronic Marketplaces", *Manage. Sci.*, Vol. 43, pp. 1676–1692, 1997.
10. Lee, H.-G., "Do Electronic Marketplaces Lower the Price of Goods?", *Commun. ACM*, Vol. 41, pp. 73–80, 1998.
11. Garicano, L. and S. N. Kaplan, "The Effects of Business-to-Business E-Commerce on Transaction Costs", *The Journal of Industrial Economics*, Vol. 49, No. 4, pp. 463–485, 2001.
12. Erik Brynjolfsson, Y. J. H. and M. D. Smith, "Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers", *Management Science*, Vol. 49, pp. 1580–1596, 2003.

13. Malone, T. W., J. Yates and R. I. Benjamin, "Electronic Markets and Electronic Hierarchies", *Commun. ACM*, Vol. 30, pp. 484–497, 1987.

14. Bailey, J. P. and Y. Bakos, "An Exploratory Study of the Emerging Role of Electronic Intermediaries", *Int. J. Electron. Commerce*, Vol. 1, pp. 7–20, 1997.

15. Choudhury, V., K. S. Hartzel and B. R. Konsynski, "Uses and Consequences of Electronic Markets: An Empirical Investigation in the Aircraft Parts Industry", *MIS Q.*, Vol. 22, pp. 471–507, 1998.

16. Glassberg, B. C., "Electronic Markets Hypothesis Redux: Where are We Now?", *Commun. ACM*, Vol. 50, pp. 51–55, 2007.

17. Grover, V., J. Lim and R. Ayyagari, "The Dark Side of Information and Market Efficiency in E-Markets*", *Decision Sciences*, Vol. 37, No. 3, pp. 297–324, 2006.

18. Feldman, S., "Electronic Marketplaces", *IEEE Internet Computing*, Vol. 4, pp. 93–95, 2000.

19. Özer, A. H. and C. Özturan, "A Model and Heuristic Algorithms for Multi-Unit Nondiscriminatory Combinatorial Auction", *Computers and Operations Research*, Vol. 36, No. 1, pp. 196–208, 2009.

20. Özer, A. H. and C. Özturan, "Multi-Unit Differential Auction-Barter Model for Electronic Marketplaces", *Electronic Commerce Research and Applications*, Vol. 10, No. 2, pp. 132–143, 2011.

21. Özer, A. H. and C. Özturan, "A Direct Barter Model for Course Add/Drop Process", *Discrete Applied Mathematics*, Vol. 159, No. 8, pp. 812–825, 2011.

22. Wang, S., S. Zheng, L. Xu, D. Li and H. Meng, "A Literature Review of Electronic Marketplace Research: Themes, Theories and an Integrative Framework", *Information Systems Frontiers*, Vol. 10, pp. 555–571, 2008.

23. Standing, S., C. Standing and P. E. D. Love, "A Review of Research on E-Marketplaces 1997-2008", *Decis. Support Syst.*, Vol. 49, pp. 41–51, 2010.

24. Bichler, M., *The Future of Emarkets: Multi-Dimensional Market Mechanisms*, Cambridge University Press, New York, NY, USA, 2001.

25. McAfee, R. P. and J. McMillan, “Auctions and Bidding”, *Journal of Economic Literature*, Vol. 25, No. 2, pp. 699–738, 1987.

26. Cassady, R., *Auctions and Auctioneering*, University of California Press, Berkeley, CA, USA, 1980.

27. Parsons, S., J. A. Rodriguez-Aguilar and M. Klein, “Auctions and Bidding: A Guide for Computer Scientists”, *ACM Comput. Surv.*, Vol. 43, pp. 10:1–10:59, 2011.

28. Wurman, P. R., W. E. Walsh and M. P. Wellman, “Flexible Double Auctions for Electronic Commerce: Theory and Implementation”, *Decision Support Systems*, Vol. 24, No. 1, pp. 17 – 27, 1998.

29. Anandalingam, G., R. W. Day and S. Raghavan, “The Landscape of Electronic Market Design”, *Management Science*, Vol. 51, No. 3, pp. 316–327, 2005, special Issue on Electronic Markets.

30. Vickrey, W., “Counterspeculation, Auctions, and Competitive Sealed Tenders”, *The Journal of Finance*, Vol. 16, No. 1, pp. 8–37, 1961.

31. Krishna, V., *Auction Theory*, Academic Press, San Diego, CA, USA, 2002.

32. Fujishima, Y., K. Leyton-Brown and Y. Shoham, “Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approaches”, *IJCAI ’99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 548–553, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

33. Sandholm, T., “An Algorithm for Optimal Winner Determination in Combinatorial Auctions”, *IJCAI ’99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 542–547, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

34. de Vries, S. and R. V. Vohra, "Combinatorial Auctions: A Survey", *INFORMS Journal on Computing*, Vol. 15, No. 3, pp. 284–309, 2003.

35. Cramton, P., Y. Shoham and R. Steinberg (Editors), *Combinatorial Auctions*, MIT Press, Cambridge, MA, USA, 2006.

36. Jackson, C., *Technology for Spectrum Markets*, Ph.D. thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, 1976.

37. Rassenti, S. J., V. L. Smith and R. L. Bulfin, "A Combinatorial Auction Mechanism for Airport Time Slot Allocation", *The Bell Journal of Economics*, Vol. 13, No. 2, pp. 402–417, 1982.

38. Caplice, C. G., *An Optimization Based Bidding Process: A New Framework for Shipper-Carrier Relationships*, Ph.D. thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.

39. Ledyard, J. O., M. Olson, D. Porter, J. A. Swanson and D. P. Torma, "The First Use of a Combined-Value Auction for Transportation Services", *Interfaces*, Vol. 32, No. 5, pp. 4–12, 2002.

40. Sheffi, Y., "Combinatorial Auctions in the Procurement of Transportation Services", *Interfaces*, Vol. 34, No. 4, pp. 245–252, 2004.

41. Graves, R. L., L. Schrage and J. K. Sankaran, "An Auction Method for Course Registration", *Interfaces*, Vol. 23, No. 5, pp. 81–92, 1993.

42. Jones, J. L., *Incompletely Specified Combinatorial Auction: An Alternative Allocation Mechanism For Business-To-Business Negotiations*, Ph.D. thesis, Warrington College of Business Administration, Univ. of Florida, Gainesville, FL, USA, 2000.

43. *The Federal Communications Commission Website*, `http://www.fcc.gov`, May 2011.

44. McMillan, J., “Selling Spectrum Rights”, *Journal of Economic Perspectives*, Vol. 8, No. 3, pp. 145–162, 1994.

45. Milgrom, P., “Putting Auction Theory to Work: The Simultaneous Ascending Auction”, *The Journal of Political Economy*, Vol. 108, No. 2, pp. 245–272, 2000.

46. Cramton, P., “Spectrum Auctions”, M. Cave, S. Majumdar and I. Vogelsang (Editors), *Handbook of Telecommunications Economics*, chap. 14, pp. 605–639, Elsevier Science, Amsterdam, The Netherlands, 2002.

47. J. O. Goeree, A. C. H. and J. O. Ledyard, *An Experimental Comparison of the FCC's Combinatorial and Non-Combinatorial Simultaneous Multiple Round Auctions*, Tech. rep., The Wireless Telecommunications Bureau of the Federal Communications Commission, 2006.

48. Rothkopf, M. H., A. Pekeč and R. M. Harstad, “Computationally Manageable Combinational Auctions”, *Management Science*, Vol. 44, No. 8, pp. 1131–1147, 1998.

49. Karp, R., “Reducibility Among Combinatorial Problems”, R. Miller and J. Thatcher (Editors), *Complexity of Computer Computations*, pp. 85–103, Plenum Press, New York, NY, USA, 1972.

50. Garey, M. R. and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, WH Freeman and Co, San Francisco, CA, USA, 1979.

51. Andersson, A., M. Tenhunen and F. Ygge, “Integer Programming for Combinatorial Auction Winner Determination”, *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pp. 39–46, IEEE Computer Society, Washington, DC, USA, 2000.

52. Sandholm, T. and S. Suri, “BOB: Improved Winner Determination in Combinatorial Auctions and Generalizations”, *Artificial Intelligence*, Vol. 145, No. 1-2, pp. 33–58, 2003.

53. Günlük, O., L. Ladànyi and S. de Vries, “A Branch-and-Price Algorithm and New

Test Problems for Spectrum Auctions", *Management Science*, Vol. 51, No. 3, pp. 391–406, 2005, special Issue on Electronic Markets.

54. Sandholm, T., A. G. S. Suri and D. Levine, "CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions", *Management Science*, Vol. 51, No. 3, pp. 374–390, 2005, special Issue on Electronic Markets.

55. *IBM ILOG CPLEX*, `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`, May 2011.

56. Leyton-Brown, K., Y. Shoham and M. Tennenholtz, "An Algorithm for Multi-Unit Combinatorial Auctions", *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 56–61, AAAI Press / The MIT Press, California, CA, USA, 2000.

57. Gonen, R. and D. Lehmann, "Optimal Solutions for Multi-Unit Combinatorial Auctions: Branch and Bound Heuristics", *EC '00: Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp. 13–20, ACM Press, New York, NY, USA, 2000.

58. Sandholm, T., S. Suri, A. Gilpin and D. Levine, "Winner Determination in Combinatorial Auction Generalizations", *AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 69–76, ACM Press, New York, NY, USA, 2002.

59. Bartal, Y., R. Gonen and N. Nisan, "Incentive Compatible Multi Unit Combinatorial Auctions", *TARK '03: Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 72–87, ACM Press, New York, NY, USA, 2003.

60. Özer, A. H., *Combinatorial Auction Based Resource Co-allocation Model for Grids*, Master's thesis, Boğaziçi University, Istanbul, Turkey, 2004.

61. Smith, V. L., "An Experimental Study of Competitive Market Behavior", *The Journal of Political Economy*, Vol. 70, No. 2, pp. 111–137, 1962.

62. Davis, D. D. and C. A. Holt, *Experimental Economics*, Princeton University Press, Princeton, NJ, USA, 1993.

63. Holt, C. A., *Markets, Games, & Strategic Behavior*, Addison-Wesley, USA, 2006.

64. Smith, V. L., "Microeconomic Systems as an Experimental Science", *The American Economic Review*, Vol. 72, No. 5, pp. 923–955, 1982.

65. Plott, C. R., "Industrial Organization Theory and Experimental Economics", *Journal of Economic Literature*, Vol. 20, No. 4, pp. 1485–1527, 1982.

66. Friedman, D., "On the Efficiency of Experimental Double Auction Markets", *The American Economic Review*, Vol. 74, No. 1, pp. 60–72, 1984.

67. Plott, C. R. and P. Gray, "The Multiple Unit Double Auction", *Journal of Economic Behavior & Organization*, Vol. 13, No. 2, pp. 245–258, 1990.

68. Huang, P., A. Scheller-Wolf and K. Sycara, "Design of a Multi-Unit Double Auction E-Market", *Computational Intelligence*, Vol. 18, No. 4, pp. 596–617, 2002.

69. Kalagnanam, J. R., A. J. Davenport and H. S. Lee, "Computational Aspects of Clearing Continuous Call Double Auctions with Assignment Constraints and Indivisible Demand", *Electronic Commerce Research*, Vol. 1, No. 3, pp. 221–238, 2001.

70. Fan, M., J. Stallaert and A. B. Whinston, "A Web-Based Financial Trading System", *Computer*, Vol. 32, No. 4, pp. 64–70, Apr 1999.

71. Xia, M., J. Stallaert and A. B. Whinston, "Solving the Combinatorial Double Auction Problem", *European Journal of Operational Research*, Vol. 164, No. 1, pp. 239–251, 2005.

72. Satterthwaite, M. A. and S. R. Williams, "The Bayesian Theory of the k-Double Auction", D. Friedman and J. Rust (Editors), *The Double Auction Market: Institutions, Theories and Evidence*, pp. 99–123, Addison-Wesley, USA, 1993.

73. Friedman, D., "The Double Auction Market Institution: A Survey", D. Friedman and J. Rust (Editors), *The Double Auction Market: Institutions, Theories and Evidence*, pp. 3–25, Addison-Wesley, USA, 1993.

74. Bao, S. and P. R. Wurman, "A Comparison of Two Algorithms for Multi-Unit k-Double Auctions", *Proceedings of the 5Th International Conference on Electronic Commerce*, ICEC '03, pp. 47–52, ACM, New York, NY, USA, 2003.

75. Chatterjee, K. and W. Samuelson, "Bargaining under Incomplete Information", *Operations Research*, Vol. 31, No. 5, pp. 835–851, 1983.

76. Satterthwaite, M. A. and S. R. Williams, "Bilateral Trade with the Sealed Bid k-Double Auction: Existence and Efficiency", *Journal of Economic Theory*, Vol. 48, No. 1, pp. 107 – 133, 1989.

77. Gode, D. K. and S. Sunder, "Lower Bounds for Efficiency of Surplus Extraction in Double Auctions", D. Friedman and J. Rust (Editors), *The Double Auction Market: Institutions, Theories and Evidence*, pp. 199–214, Addison-Wesley, USA, 1993.

78. Parsons, S., M. Marcinkiewicz, J. Niu and S. Phelps, *Everything You Wanted to Know About Double Auctions but were Afraid to (Bid or) Ask*, Tech. rep., Department of Computer and Information Science, Brooklyn College, City University of New York, 2006.

79. Phelps, S., S. Parsons and P. McBurney, "An Evolutionary Game-Theoretic Comparison of Two Double-Auction Market Designs", P. Faratin and J. Rodrguez-Aguilar (Editors), *Agent-Mediated Electronic Commerce VI. Theories for and Engineering of Distributed Mechanisms and Systems*, Vol. 3435 of *Lecture Notes in Computer Science*, pp. 101–114, Springer Berlin / Heidelberg, 2005.

80. Jevons, W. S., *The Theory of Political Economy*, Macmillan and Co., London, 1888.

81. Özturan, C., "Resource Bartering in Data Grids", *Scientific Programming*, Vol. 12, No. 3, pp. 155–168, 2004.

82. Özturan, C., "Network Flow Models for Electronic Barter Exchanges", *Journal of Organizational Computing and Electronic Commerce*, Vol. 14, No. 3, pp. 175–194, 2004.

83. Gutin, G., “Finding a Longest Path in a Complete Multipartite Digraph”, *SIAM J. Discret. Math.*, Vol. 6, pp. 270–273, 1993.

84. Bang-Jensen, J. and G. Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Publishing Company, Incorporated, London, UK, 2008.

85. López, N., M. Núñez, I. Rodríguez and F. Rubio, “A Formal Framework for E-Barter Based on Microeconomic Theory and Process Algebras”, H. Unger, T. Bhme and A. Mikler (Editors), *Innovative Internet Computing Systems*, Vol. 2346 of *Lecture Notes in Computer Science*, pp. 217–228, Springer Berlin / Heidelberg, 2002.

86. López, N., M. Núñez, I. Rodríguez and F. Rubio, “A Multi-Agent System for E-Barter Including Transaction and Shipping Costs”, *Proceedings of the 2003 ACM symposium on Applied computing*, SAC ’03, pp. 587–594, ACM, New York, NY, USA, 2003.

87. Buttyán, L., L. Dóra, M. Félegyházi and I. Vajda, “Barter Trade Improves Message Delivery in Opportunistic Networks”, *Ad Hoc Networks*, Vol. 8, No. 1, pp. 1 – 14, 2010.

88. Cohen, B., *Incentives Build Robustness in BitTorrent*, 2003, `http://www.bittorrent.org/bittorrentecon.pdf`, May 2011.

89. *eMule-Project.net*, `http://www.emule-project.net`, May 2011.

90. Aperjis, C., M. J. Freedman and R. Johari, “Peer-Assisted Content Distribution with Prices”, *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT ’08, pp. 17:1–17:12, ACM, New York, NY, USA, 2008.

91. Menasche, D. S., L. Massoulie and D. Towsley, “Reciprocity and Barter in Peer-to-Peer Systems”, *INFOCOM ’10: IEEE International Conference on Computer Communications*, pp. 1–9, IEEE Computer Society, Washington, DC, USA, 2010.

92. Alvin E. Roth, T. S. and M. U. Ünver, “Efficient Kidney Exchange: Coincidence of Wants in Markets with Compatibility-Based Preferences”, *The American Economic Review*, Vol. 97, pp. 828–851, 2007.

93. Alvin E. Roth, T. S. and M. U. Ünver, "Kidney Exchange", *The Quarterly Journal of Economics*, Vol. 119, pp. 457–488, 2004.

94. Saidman, S., A. Roth, T. Sönmez, M. Ünver and F. Delmonico, "Increasing the Opportunity of Live Kidney Donation by Matching for Two- and Three-Way Exchanges", *Transplantation*, Vol. 81, pp. 773–782, 2006.

95. Abraham, D. J., A. Blum and T. Sandholm, "Clearing Algorithms for Barter Exchange Markets: Enabling Nationwide Kidney Exchanges", *Proceedings of the 8th ACM conference on Electronic commerce*, EC '07, pp. 295–304, ACM, New York, NY, USA, 2007.

96. Haddawy, P., N. Rujikeadkumjorn, K. Dhananaiyapergse and C. Cheng, "Balanced Matching of Buyers and Sellers in E-Marketplaces: The Barter Trade Exchange Model", *ICEC*, pp. 85–94, 2004.

97. *The International Reciprocal Trade Association*, `http://www.irta.com`, May 2011.

98. Burke, E. K. and S. Petrovic, "Recent Research Directions in Automated Timetabling", *European Journal of Operational Research*, Vol. 140, No. 2, pp. 266–280, 2002.

99. Carter, M. W. and G. Laporte, "Recent Developments in Practical Course Timetabling", *Practice and Theory of Automated Timetabling II*, Vol. 1408 of *Lecture Notes in Computer Science*, pp. 3–19, Springer-Verlag, Berlin, 1998.

100. Lewis, R., "A Survey of Metaheuristic-Based Techniques for University Timetabling Problems", *OR Spectrum*, Vol. 30, pp. 167–190, 2008.

101. Schaerf, A., "A Survey of Automated Timetabling", *Artificial Intelligence Review*, Vol. 13, No. 2, pp. 87–127, 1999.

102. *Boğaziçi University Online Registration System*, `http://registration.boun.edu.tr`, May 2011.

103. Müller, T. and K. Murray, “Comprehensive Approach to Student Sectioning”, *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling*, 2008.

104. *UniTime*, `http://www.unitime.org`, May 2011.

105. Özturan, C., “Used Car Salesman Problem: A Differential Auction–Barter Market”, *Annals of Mathematics and Artificial Intelligence*, Vol. 44, pp. 255–267, 2005.

106. Ahuja, R. K., T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, New Jersey, NJ, USA, 1993.

107. Goldberg, A. V. and R. E. Tarjan, “Finding Minimum-Cost Circulations by Canceling Negative Cycles”, *J. ACM*, Vol. 36, No. 4, pp. 873–886, 1989.

108. *Software and Supplementary Materials for A Direct Barter Model for Course Add/Drop Process*, `http://www.ahozer.com/en/pubs/adddrop`, May 2011.

109. *Andrew Goldberg's Network Optimization Library*, `http://www.avglab.com/andrew/soft.html`, May 2011.

110. Goldberg, A. V., “An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm”, *J. Algorithms*, Vol. 22, No. 1, pp. 1–29, 1997.

111. Al-Yakoob, S. M. and H. D. Sherali, “Mathematical Programming Models and Algorithms for a Class-Faculty Assignment Problem”, *European Journal of Operational Research*, Vol. 173, No. 2, pp. 488–507, 2006.

112. Dimopoulou, M. and P. Miliotis, “Implementation of a University Course and Examination Timetabling System”, *European Journal of Operational Research*, Vol. 130, No. 1, pp. 202–213, 2001.

113. Ismayilova, N. A., M. Sağır and R. N. Gasimov, “A Multiobjective Faculty-Course-Time Slot Assignment Problem with Preferences”, *Mathematical and Computer Modelling*, Vol. 46, No. 7-8, pp. 1017–1029, 2007.

114. Hertz, A., "Finding a Feasible Course Schedule Using Tabu Search", *Discrete Applied Mathematics*, Vol. 35, No. 3, pp. 255 – 270, 1992.

115. Reeves, G. R. and E. P. Hickman, "Assigning MBA Students to Field-Study Project Teams - A Multicriteria Approach", *Interfaces*, Vol. 22, No. 5, pp. 52–58, 1992.

116. Willoughby, K. A. and C. J. Zappe, "A Methodology to Optimize Foundation Seminar Assignments", *Journal of the Operational Research Society*, Vol. 57, No. 8, pp. 950–956, 2006.

117. Alvarez-Valdes, R., E. Crespo and J. M. Tamarit, "Assigning Students to Course Sections Using Tabu Search", *Annals of Operations Research*, Vol. 96, No. 1-4, pp. 1–16, 2000.

118. Hertz, A. and V. Robert, "Constructing a Course Schedule by Solving a Series of Assignment Type Problems", *European Journal of Operational Research*, Vol. 108, No. 3, pp. 585–603, 1998.

119. Sampson, S. E., J. R. Freeland and E. N. Weiss, "Class Scheduling to Maximize Participant Satisfaction", *Interfaces*, Vol. 25, No. 3, pp. 30–41, 1995.

120. Hinkin, T. R. and G. M. Thompson, "SchedulExpert: Scheduling Courses in the Cornell University School of Hotel Administration", *Interfaces*, Vol. 32, No. 6, pp. 45–57, 2002.

121. Gale, D. and L. S. Shapley, "College Admissions and the Stability of Marriage", *The American Mathematical Monthly*, Vol. 69, No. 1, pp. 9–15, 1962.

122. Baïou, M. and M. Balinski, "Erratum: The Stable Allocation (or Ordinal Transportation) Problem", *Mathematics of Operations Research*, Vol. 27, No. 4, pp. 662–680, 2002.

123. Kellerer, H., U. Pferschy and D. Pisinger, *Knapsack Problems*, Springer, Berlin, Germany, 2004.

124. Dumas, M., B. Benatallah, N. Russell and M. Spork, "A Configurable Matchmaking Framework for Electronic Marketplaces", *Electronic Commerce Research and Applications*, Vol. 3, No. 1, pp. 95–106, 2004.

125. Ono, C., S. Nishiyama and H. Horiuchi, "Reducing Complexity in Winner Determination for Combinatorial Ascending Auctions", *Electronic Commerce Research and Applications*, Vol. 2, No. 2, pp. 176–186, 2003.

126. Mu'alem, A. and N. Nisan, "Truthful Approximation Mechanisms for Restricted Combinatorial Auctions", *Games and Economic Behavior*, Vol. 64, No. 2, pp. 612–631, 2008.

127. Fudenberg, D. and J. Tirole, *Game Theory*, The MIT Press, USA, 1991.

128. *National Auto Auction Association's Annual Review*, `http://www.naaamap.com/NAAA/annual_review/2009AnnualReview.pdf?pageid=3372`, May 2011.

129. Haddawy, P., C. Cheng, N. Rujikeadkumjorn and K. Dhananaiyapergse, "Optimizing Ad Hoc Trade in a Commercial Barter Trade Exchange", *Electronic Commerce Research and Applications*, Vol. 4, No. 4, pp. 299–314, 2005.

130. *Software and Supplementary Materials for the Multi-Unit Differential Auction-Barter Model for Electronic Marketplaces*, `http://www.ahozer.com/en/pubs/mudab`, May 2011.

131. *Global Reciprocal Trade Statistics*, `http://www.irta.com/component/content/article/39.html`, May 2011.

132. Wyatt, E., *Internet Grows as Factor in Used-Book Business*, 2005, `http://www.nytimes.com/2005/09/29/books/29book.html`, May 2011.

133. Kane, Y. I. and M. Bustillo, *Used Games Score Big for GameStop*, 2009, `http://online.wsj.com/article/SB123249378212700025.html`, May 2011.

134. Smith, M. D. and R. Telang, "Internet Exchanges for Used Digital Goods", *SSRN eLibrary*, 2008.

135. Ghose, A., M. D. Smith and R. Telang, "Internet Exchanges for Used Books: An Empirical Analysis of Product Cannibalization and Welfare Impact", *Info. Sys. Research*, Vol. 17, pp. 3–19, 2006.

136. Arunkundram, R. and A. Sundararajan, "An Economic Analysis of Electronic Secondary Markets: Installed Base, Technology, Durability and Firm Profitability", *Decis. Support Syst.*, Vol. 24, pp. 3–16, 1998.

137. Ghose, A., R. Telang and R. Krishnan, "Effect of Electronic Secondary Markets on the Supply Chain", *J. Manage. Inf. Syst.*, Vol. 22, pp. 91–120, 2005.

138. *GNU Scientific Library*, `http://www.gnu.org/software/gsl`, May 2011.

139. Ghose, A., "Internet Exchanges for Used Goods: An Empirical Analysis of Trade Patterns and Adverse Selection", *MIS Quarterly*, Vol. 33, pp. 263–291, 2009.

140. *Software and Supplementary Materials for the Double Auction with Limited Cover Money Model*, `http://www.ahozer.com/en/pubs/dalcom`, May 2011.