A Double Auction Based Mathematical Market Model and Heuristics for Internet-Based Secondhand Durable Good Markets

Ali Haydar Özer*

Department of Computer Engineering,

Marmara University, Istanbul, Turkey.

E-mail address: *haydar.ozer@marmara.edu.tr*

This is the accepted version of the article submitted to Computers & Operations Research journal. This version contains all the materials without professional editing. The published journal article is available at `https://doi.org/10.1016/j.cor.2019.06.005`

Full Citation:

Ali Haydar Özer,

A double auction based mathematical market model and heuristics for internet-based secondhand durable good markets,

Computers & Operations Research,

Volume 111,

2019,

Pages 116-129,

ISSN 0305-0548,

https://doi.org/10.1016/j.cor.2019.06.005.

*Address: Department of Computer Engineering, Faculty of Engineering, Marmara University, 34722, Istanbul, Turkey. Tel: +90-216-348-0292 x1247 Fax: +90-216-347-2859 E-mail address: haydar.ozer@marmara.edu.tr

A Double Auction Based Mathematical Market Model and Heuristics for Internet-Based Secondhand Durable Good Markets

Ali Haydar Özer*

Department of Computer Engineering,

Marmara University, Istanbul, Turkey.

E-mail address: *haydar.ozer@marmara.edu.tr*

*Address: Department of Computer Engineering, Faculty of Engineering, Marmara University, 34722, Istanbul, Turkey. Tel: +90-216-348-0292 x1247 Fax: +90-216-347-2859 E-mail address: haydar.ozer@marmara.edu.tr

Abstract

Trading of used goods in secondhand markets is considered to have both environmental and economic benefits. For the trading of used goods as well as new ones, this study proposes a periodic double auction model for Internet-based electronic markets whereby each bidder can place both asks and bids for selling and purchasing goods respectively during a trading period. The proposed model allows each bidder to place a spending limit order such that the difference between the cost of the purchased goods and the income obtained from the sold goods does not exceed this limit. Furthermore, since a bidder may be indifferent to multiple goods, e.g. copies of a same title, the model also provides a mechanism so that bidders may combine a number of bids inside a set and put a limit on the number of goods to be purchased in this set. The model is mathematically defined, and the corresponding winner determination problem is formulated using linear integer programming. Since the problem is shown to be NP-hard, a number of heuristic methods are also proposed. Performances of these methods are evaluated on a comprehensive test suite and statistical analyses of the results are presented. Furthermore, possible economical contribution of the model is also evaluated. The results indicate that the proposed model can be used efficiently in large-scale markets with tens of thousands of bidders.
Keywords : Electronic Market; Double Auction; Budget Constraint; Used Good Trading; Secondhand Market; Market Design; Mathematical Modeling; Integer Programming; Combinatorial Optimization.

1 Introduction

Although introduced in late 1980's (Pearce and Turner, 1989), the concept of the Circular Economy has recently received much attention as a way to reduce the environmental burden of current unsustainable production and consumption model (see Ghisellini et al. (2016) for a review). The Circular Economy aims to close the loops in industrial ecosystems with the well-known slogan "reduce, reuse, recycle" (Stahel, 2016; Thomas, 2011). It has been promoted by environmental agencies such as the European Environment Agency (The European Commission, 2015) and the U.S. Environmental Protection Agency (United States Environmental Protection Agency, 2017), and Germany, Japan and China have already integrated this concept into their laws (Geissdoerfer et al., 2017; Yuan et al., 2006).

Within the concept of the Circular Economy, a key aspect of reuse strategy is the trading of used durable goods. Trading of a durable good through a secondhand market extends the life span of the good, and hence is considered to have positive environmental impacts as long as the amount of resources required for the usage phase of the good is smaller than the resources required for new production (Clausen et al., 2010; Cooper and Gutowski, 2017). This is the case for many types of goods that consume small amount of energy or no energy at all such as books, furniture, and consumer electronics (Clausen et al., 2010).

Aside from the environmental benefits, used goods trading has also economic benefits. From the buyers' perspective, the presence of secondhand markets always increases buyers' welfare (Arunkundram and Sundararajan, 1998). These benefits also provide incentive for market makers to build and execute such markets, the well-known of examples of which are eBay and Amazon.com. Ghose et al. (2006) estimate that Amazon.com obtains a profit of approximately \$65 million from used book sales. The authors also estimate an increase of approximately \$88 million in the total welfare. In line with this estimation, Thomas (2003) argues that secondhand markets always increase overall economic welfare independent of whether they affect the demand for new goods.

Considering these environmental and economic benefits, widespread trading of used goods should be favored. Accordingly, in order to provide development or growth of secondhand markets, this study introduces a market model called Double Auction with Budget Constraints (DABC) model for Internet-based secondhand markets. The DABC model utilizes the periodic (discrete-time) double auction (DA) institution for the trading of used goods as well as new ones (the DA institution is described in Section 2). Market participants, i.e. bidders, can have both buyer and seller roles, and therefore each bidder can submit both *asks*, i.e. sales orders for the goods they want to sell, and *bids*, i.e. purchase orders for the goods they want to purchase inside the same trading period. For pricing the goods, the well-known k-DA policy is used. The model keeps track of the budgets of the bidders so that each bidder can spend the income to be obtained from her goods for the goods she wants to buy. The model also features a spending limit mechanism such that for each bidder the difference between the expenses and the income does not exceed the declared spending limit of the bidder. This mechanism enables bidders to participate in the market without the risk of having a budget deficit.

Another issue handled in the proposed model is that depending on her preferences, a bidder may treat a number of goods as substitutable or indistinguishable. For instance, a bidder may be indifferent to multiple copies of the second edition of a textbook offered in the market the quality of which ranges from good to excellent. Another may request the same textbook without distinguishing its edition or quality. The DABC model further provides a mechanism for handling such preferences of the bidders. Each bidder is allowed to combine multiple offers inside a single bid and restrict the number of goods to be purchased, for instance to only one, in that bid. Of course, she is also able to offer different prices for the goods listed in the bid through which she is able to encode her preference for each of these goods.

By means of these features, the DABC model aims *to attract more participants* to the used good markets and *to increase the total economic welfare* of the participants.

In the next section, a brief overview of the DA institution is given. In Section 3, the DABC model is explained in detail on an example market scenario. Section 4 introduces the mathematical definition of the model. The linear integer programming formulation of the winner determination problem of the model is also presented. It is proven that the decision version of the winner determination problem is NP-complete, and it is inapproximable unless $P = NP$. Since the problem is intractable, and a number of heuristic methods are also proposed. Section 5 introduces these methods in detail. In order to evaluate the performances of these methods, a number of experiments are conducted, and the results of these experiments are presented in Section 6. Finally, the paper is concluded in Section 7.

2 A Brief Review of Double Auction Institution

As defined by Friedman (1993), a *market institution* is "a set of rules specifying which sorts of bids and other messages are legitimate, and how and when specific traders transact, given their chosen messages". Among the market institutions, the double auction (DA) institution is an important and widely used institution especially in stock exchanges. In DA institution, multiple potential sellers and buyers coexist in the auction and can submit multiple *asks* (sales orders) and *bids* (purchase orders) *simultaneously* to the auctioneer for well-defined commodities. Then, the auctioneer determines a *clearing price* p for each commodity. The sellers who asked less than or equal to p sells their commodities and the buyers who bid greater than or equal to p purchase the corresponding commodities at the clearing price p.

Beginning with Smith (1962), many laboratory experiments with DA rules have been carried out (see Davis and Holt (1993) and Holt (2006) for a survey). The experiments demonstrate that the DA institution results in high allocative efficiency even with a small number of traders (Friedman, 1984; Plott, 1982; Smith, 1982) 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 good. The *multi-unit DA* (Huang et al., 2002; Plott and Gray, 1990) is an extension to the single unit DA which allows multiple instances of a commodity to be traded in the auction. Kalagnanam et al. (2001) 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* (Fan et al., 1999; Xia et al., 2005), the institution further allows package (combinatorial) bidding in which the participants can submit bids on bundles of goods instead of a single good as in combinatorial auctions.

According to the market clearing interval, there are two types of the DA institution. In the *continuous time* DA institution, asks and bids can be submitted and retracted any time during the auction, and exchanges of goods 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 stock exchanges.

In the *discrete time* DA institution, which is also known as the *call-market* (Satterthwaite and Williams, 1993) or the *clearinghouse* (Friedman and Rust, 1993), 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 (Bao and Wurman, 2003; Wurman et al., 1998).

For determining the market equilibrium prices in a discrete time DA institution, k-DA policy can be used (Chatterjee and Samuelson, 1983). 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 good (Satterthwaite and Williams, 1989).

In terms of allocative efficiency, the expected efficiency of a discrete time DA institution is higher than that of the continuous time DA (Gode and Sunder, 1993; Parsons et al., 2006). 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 (Phelps et al., 2005).

For further details on DA institution, the reader is referred to the surveys (Friedman and Rust, 1993; Parsons et al., 2006).

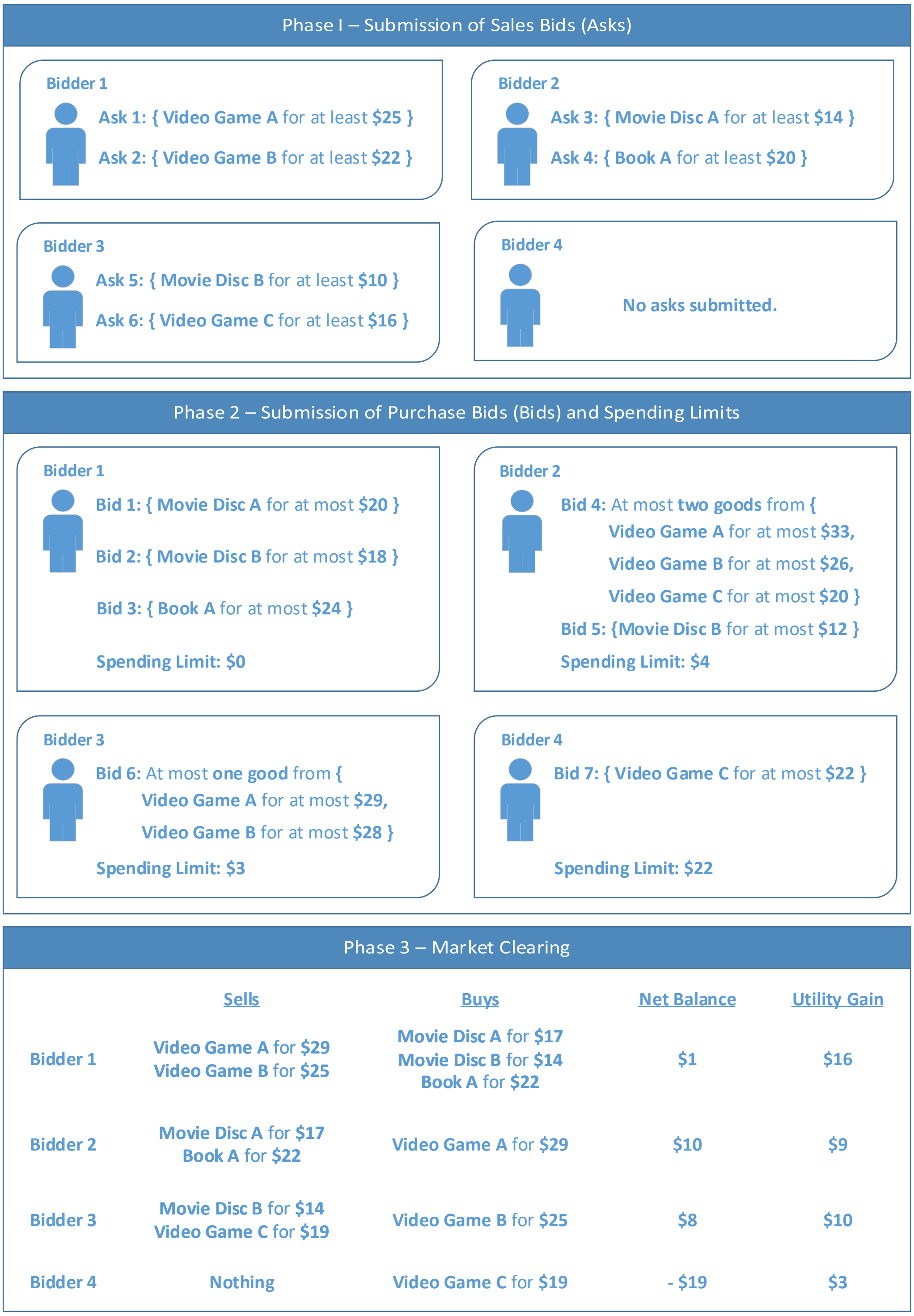

Figure 1: Example market scenario for demonstrating the DABC model.

3 The DABC Model

In this section, the DABC model will be introduced through an example market scenario which is illustrated in Figure 1.

In this model, each bidder can sell and purchase goods simultaneously, i.e. bidders can have both buyer and seller roles inside a single trading round. Firstly, bidders declare the goods they want to sell along with a price they request for each good, i.e. submit asks to the auctioneer. For instance, in this scenario, Bidder 1 wants to sell two video games, Video Games A and B and asks for \$25 and \$22 for these two games respectively. These prices are the ask prices. An *ask price* indicates the *minimum* amount of money the bidder is willing to get if the corresponding good is sold. However, the actual transaction prices could be higher. Similarly, Bidder 2 asks \$14 for Movie Disc A and \$20 for Book A, and Bidder 3 asks \$10 for Movie Disc B and \$16 for Video Game C. Bidder 4, on the other hand, submits no asks, i.e. she has only a buyer role.

Secondly, the bidders declare their purchase bids as shown in Figure 1. In the DABC model, each instance of a good in the market is considered as a unique good since the instances of a same good can be differentiated by the bidders. For instance, suppose that there are multiple copies of a same book to be sold in the market, however:

- conditions of the copies may be different such as like-new, acceptable, has some noticeable wears;
- there may be several editions of the same title;
- reputations of the sellers of the copies may be different;
- locations of the sellers, and therefore the associated shipment costs and the corresponding environmental impacts may vary.

These differences between the copies of a same title could be important for some bidders and not for the others. Therefore, in this model, each bidder is allowed to encode her preferences of substitutabilities in her bids.

This feature can be seen in the example bids. For instance, in Bid 6, Bidder 3 considers that Video Game A and Video Game B are substitutable and declares that she wants to purchase one of these games. She bids \$29 and \$28 for these two games respectively. These prices are the bid prices. A *bid price* indicates the *maximum* amount of money the bidder is willing to spend for the corresponding good. However, the actual transaction prices could be lower.

Bidder 2, on the other hand, want to purchase at most two games among the video games A, B and C. Thus, in Bid 4, she bids \$33, \$26 and \$20 for these three games respectively.

However, Bidder 2 has a further request. She also wants to purchase movie disc B (a Blu-Ray disc for example) for at most \$12. Therefore, she submits a second bid (Bid 5) containing only one good. When the market is cleared, this bidder may get up to three goods.

After the bidders submit their purchase bids, each bidder with a buyer role declares a spending limit which indicates the maximum amount the bidder may spend in the market. If a bidder has also goods for sale, the model adds the income from the sold goods to the budget of the bidder. In this case, the spending limit indicates the bound on the difference between the expenses and the income of the bidder. In other words, the amount spent on the purchased goods minus the income obtained from the sold goods cannot exceed this limit value. For instance, Bidder 3 wants to sell two goods for at least \$10 and \$16 and wants to purchase one of the video games A or B for \$29 or \$28 respectively. She declares a spending limit of \$3. Thus, she will have enough budget to purchase one of the video games A or B only if both of her goods are sold. However, if she had declared a spending limit of \$29 for instance, then she would be able to purchase one of these games even if none of her goods are sold. Note that these examples are given on the basis that the transactions occur at the corresponding ask and bid prices.

In the DABC model, the *utility* of a transaction between a seller and a buyer is defined as the difference between the corresponding bid and ask prices for the good to be exchanged. Negative utility values are not allowed, i.e. the transactions are not subsidized by the market maker. Therefore, bids whose bid price is smaller than the corresponding ask price are simply discarded. For pricing the goods, k-DA policy, as explained in Section 2, is used. Accordingly, the price of an exchanged good is defined as

$$k \cdot \textit{bid price} + (1 - k) \cdot \textit{ask price}$$

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. A typical value for k is 0.5 in which the utility is distributed equally. However, in real life markets, the market maker should also be paid in order to keep the market running. In the DABC model, the market maker can be paid by a share of the utility obtained in each transaction.

The objective of the DABC model is to maximize the sum of the transaction utilities so as to maximize the total economic welfare of the bidders. In order to achieve the optimum market efficiency, all the bidders should declare their true valuations for the goods in the market. However, if they can benefit, the bidders are tempted to declare their own valuations incorrectly as a strategic behavior. In this case, the efficiency of the market reduces. A market mechanism which ensures that the declaration of a bidder's true valuation is the best strategic behavior for that bidder independent of how other bidders act in the market, is called *truthful* or *incentive compatible* in dominant strategies. This pricing scheme is, on the other hand, is not incentive compatible in dominant strategies (Babaioff and Nisan, 2004). However, this theoretical limitation might not cause great loss of efficiency in practice. For instance, Wilson (1985) analyzed a sealed bid double auction market institution (a multi-trader version of the k-DA), and concluded that the market institution is *incentive efficient*, meaning that there is no incentive for bidders to prefer another institution. In addition to this result, Rustichini et al. (1994) have shown experimentally that the inefficiency of k-DA institution with only six buyers and six sellers in a market is less than 1%. Furthermore, Satterthwaite and Williams (2002) have shown that no other auction mechanism converges faster than k-DA in the worst-case. Thus, the inefficiency of k-DA diminishes rapidly as the number of bidders increases which is expected in large-scale online markets. For more detailed discussion of the efficiency of double auction mechanisms, the reader is referred to the works of Parsons et al. (2006) and Babaioff and Nisan (2004).

As noted earlier, the primary feature of the DABC model is to allow bidders to use the revenue to be obtained from the goods they sell in the market for purchasing new goods. That is, transparently directing the flow of income to the expenses. The benefit of this feature can also be seen in this scenario. As indicated by their spending limits, Bidders 1, 2 and 3, do not have enough budget to purchase the goods they want. Therefore, in traditional markets, they would have to sell their goods first, and using the income, they would be able to purchase new goods. Thus, for this particular scenario in a traditional market, only one transaction occurs. Bidder 4 buys video game C from Bidder 3 for \$19 (assuming the transaction price is the average value). After this transaction Bidder 3 would get \$19, however this amount does not allow her to purchase a good she wants even when combined with her spending limit, i.e. budget, of \$3. Thus, the trading volume of the market would be \$19, and the total utility would be just \$6. However, the outcome of the DABC model for this market scenario is quite different as seen in Figure 1. Using the DABC model, all six goods offered in the market are sold with a trading volume of \$126 and total utility of \$38, and none of the bidders spend more than their spending limits in the market.

The implementation of the model is also straightforward. A trading round consists of three periods. In the first period, asks are collected from bidders also with possible modifications during this period. In the second period, bidders submit their bids along with their spending limits. Again, the bidders may modify their bids and spending limits during this period. After the second period is over, in the third period, the market is cleared by solving the winner determination problem which will be introduced in the next section. The winning bids are, then, announced to the bidders. Unsatisfied asks/bids of a bidder could be transferred to the next bidding round in case the bidder requests so. 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 higher trading volume but reduces the market throughput.

4 Formulation of the DABC Model

The DABC model is formally defined as follows:

Let $D = \{d_1, d_2, \ldots, d_m\}$ be the set of m market participants, i.e. bidders, where a bidder may have only a seller role, only a buyer role or may have both roles. Each bidder with a seller role declares a set of goods to be sold, G_i, and the corresponding ask prices ($1 \leq i \leq m$, $G_i = \emptyset$ if d_i has only a buyer role). Since each good is considered unique in this model, the sets G_i do not intersect ($\forall i, i' \,|\, G_i \cap G'_i = \emptyset$). Furthermore, let the set $G = \{g_1, g_2, \ldots, g_n\}$ denote all the goods available in the market where $G = \bigcup_{i=1}^{m} G_i$. $\alpha(g_j)$ indicates the corresponding ask price of the good g_j. Note that the ask price $\alpha(g_j)$ indicates the minimum amount for which the owner of the good g_j is willing to sell for ($1 \leq j \leq n$, $\alpha(g_j) \in \mathbb{R}^+ \cup \{0\}$).

Each bidder with a buyer role submits a spending limit, and these limits are denoted with tuple $L = (l_1, l_2, \ldots, l_m)$ where l_i is the spending limit of the bidder d_i ($l_i \in \mathbb{R}^+ \cup \{0\}$). For bidders who do not want to put a limit on their purchases, a sufficiently high value may be defined as a limit without affecting the outcome.

A bid in the DABC model 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 elements, $R_k = \{r_{k1}, r_{k2}, \ldots, r_{kz}\}$ where r_{kl} denotes the good to be purchased, i.e. the requested good. Along with each good requested, the bidder submits a bid price and $\beta(r_{kl})$ indicates the bid price of the good r_{kl}. Note that the bid price $\beta(r_{kl})$ indicates the maximum amount for which the owner of the bid b_k is willing to purchase the good r_{kl} ($1 \leq l \leq z$, $r_{kl} \in G$, $\beta(r_{kl}) \in \mathbb{R}^+ \cup \{0\}$). The bids with an offered price less than the corresponding ask price are ignored ($\forall k, l \mid \beta(r_{kl}) \geq \alpha(r_{kl})$). The upper purchase limit, on the other hand, indicates the maximum number of goods that the bidder is willing to purchase among the goods in R_k. Finally, the set of bids submitted by the bidder d_i is denoted as B_i, and the set of all bids, $B = \{b_1, b_2, \ldots, b_v\}$, is defined as $B = \bigcup_{i=1}^{m} B_i$.

When a transaction occurs, the price of the good r_{kl} is determined as $P_{kl} = \mathtt{k} \cdot \beta(r_{kl}) + (1 - \mathtt{k}) \cdot \alpha(r_{kl})$ where $\mathtt{k} \in [0, 1]$ is the parameter of the k-DA policy. The utility of this transaction is defined as $U_{kl} = \beta(r_{kl}) - \alpha(r_{kl})$.

The bid b_k is called *satisfiable* if there exists *at least* one good in the request set R_k which is available for purchase and the price of this good is within the budget of the bidder. The model ensures that the budget of the bidder is never deficit. The budget of a bidder is defined as the proceeds of the sold goods added with the spending limit of the bidder, minus the cost of the purchased goods.

The *winner determination problem* (WDP) of the DABC model is defined as finding the maximum cardinality set of mutually satisfiable bids such that the total utility is maximized.

In order to formulate the problem using linear integer programming, a binary variable x_{kl} is introduced. It denotes whether the item r_{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:

$$\text{maximize} \sum_{\forall k,l \,|\, b_k \in B \wedge r_{kl} \in R_k} U_{kl} \cdot x_{kl} \quad (1)$$

$$\text{subject to} \sum_{\forall l \,|\, r_{kl} \in R_k} x_{kl} \leq u_k \qquad (\forall k |\, b_k \in B) \quad (2)$$

$$\sum_{\forall k,l \,|\, b_k \in B \wedge r_{kl} \in R_k \wedge r_{kl} = g_j} x_{kl} \leq 1 \qquad (\forall j \,|\, g_j \in G) \quad (3)$$

$$\sum_{\forall k,l \,|\, b_k \in B_i \wedge r_{kl} \in R_k} P_{kl}\, x_{kl} - \sum_{\forall k,l \,|\, b_k \in B \wedge r_{kl} \in R_k \wedge r_{kl} \in G_i} P_{kl}\, x_{kl} \leq l_i \qquad (\forall i \,|\, d_i \in D) \quad (4)$$

$$x_{kl} \in \{0,1\} \; (\forall k, l) \quad (5)$$

In this formulation, Eq.(1) is the objective function which maximizes the total utility obtained from all transactions. Eq.(2) enforces the upper purchase limits of the bids, i.e. for every bid, the number of goods that can be purchase is at most u_k. Eq.(3) ensures that each good is sold to at most one bidder. Finally, Eq.(4) is the budget constraint, that is for each bidder, the total cost of the purchased goods cannot exceed the sum of the spending limit of the bidder and the proceeds of the sold goods.

4.1 Theorems Related to the Winner Determination Problem

In this section, complexity results related to the WDP are presented.

Proposition 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 D, a set of goods G and associated ask prices $\alpha(g_i)$, spending limits of the bidders L, a set of bids B and a positive integer K, is there a simultaneously satisfiable subset $B' \subseteq B$ such that the total utility is greater than or equal to K?

If we have a certificate that consists of set B', this certificate can be verified in polynomial time by checking Eq.(2-4) using B'. Therefore, Π is in NP.

Next, a polynomial time transformation from the subset sum problem will be presented. Let Π' be the subset sum problem (Garey and Johnson, 1979, 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 D consist of two bidders ($D = \{d_1, d_2\}$). Let the set of goods to be sold by the bidder d_1 be $G_1 = \{g_1, g_2, \ldots, g_f\}$ and by the bidder d_2 be $G_2 = \{g_{f+1}\}$ ($f = |A|$). The ask prices of all the goods are zero ($\alpha(g_i) = 0,\ 1 \leq i \leq f + 1$). Also let the spending limits of both bidders be zero ($l_1 = 0,\ l_2 = 0$). Assume that the bidder d_1 submits only one bid requesting one good g_{f+1} with a bid price of $B/\mathtt{k}$, and the bidder d_2 also submits one bid requesting up to f goods of the bidder d_1 with the following bid prices $a_1/\mathtt{k}, a_2/\mathtt{k}, \ldots, a_f/\mathtt{k}$. That is $b_1 = (\{g_{f+1}\}, 1)$, $\beta(r_{11}) = B/\mathtt{k}$; and $b_2 = (\{g_1, g_2, \ldots, g_f\}, f)$, $\beta(r_{2l}) = a_l/\mathtt{k},\ 1 \leq l \leq f$). Finally, let $K = 2B/\mathtt{k}$.

Since the spending limits 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. □

This proof leads to the inapproximability result for the WDP.

Lemma 1. *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 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 only for a special subset of the WDP instances in which the spending limits of all the bidders are zero. The reason is that in all nontrivial feasible solutions of such an instance, for *every* bidder the total price of the sold goods should be equal to the total price of the purchased goods. Thus, in practice, most of the problem instances of this kind would possibly have no nontrivial feasible solutions at all. However, these rare situations can also be avoided by the market maker by forcing a lower limit for the spending limits of bidders.

On the other hand, if the spending limits of all the bidders are as high as to allow every bidder to purchase every good they want, the corresponding WDP instances become easier to solve. This special case is called as *budget-relaxed WDP*.

Proposition 2. *The budget-relaxed WDP is in P.*

Proof. The budget-relaxed WDP can be modeled as a minimum cost flow problem (Ahuja et al., 1993). 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 $v \in V$.

The network is constructed by introducing a node for each bid $b_k \in B$ and for each good $g_j \in G$. 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 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 ($c(s, b_k) = 0$). Then, for each bid $b_k \in B$ and for each requested good $r_{kl} \in R_k$, an arc is drawn from the node b_k to r_{kl}. The upper bound of this arc is one, that is $u(b_k, r_{kl}) = 1$, and the cost of the arc is the additive inverse of the utility value ($c(b_k, r_{kl}) = \alpha(r_{kl}) - \beta(r_{kl})$). Finally, for each good $g_j \in G$, an arc is introduced from the node g_j to the sink node t with an upper bound of one and cost of zero ($u(g_j, t) = 1$, $c(g_j, t) = 0$). 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 $v \in V$.

The constructed network is shown in Figure 2. The minimum cost flow in this network gives the optimum solution for the budget-relaxed WDP since the constraints in Eq.(2,3) are encoded in the network and minimizing the cost in the network means maximizing the total utility. □

5 Solution Methods

In Section 4, the WDP of the DABC model was defined and formulated as an integer linear program. Therefore, DABC model instances can be solved using general purpose integer programming solvers. However, since the WDP is NP-hard, general purpose solvers may not be able to find the optimal solution or any solution at all for difficult instances within a given time limit. Therefore, heuristic approaches would be necessary for solving such instances.

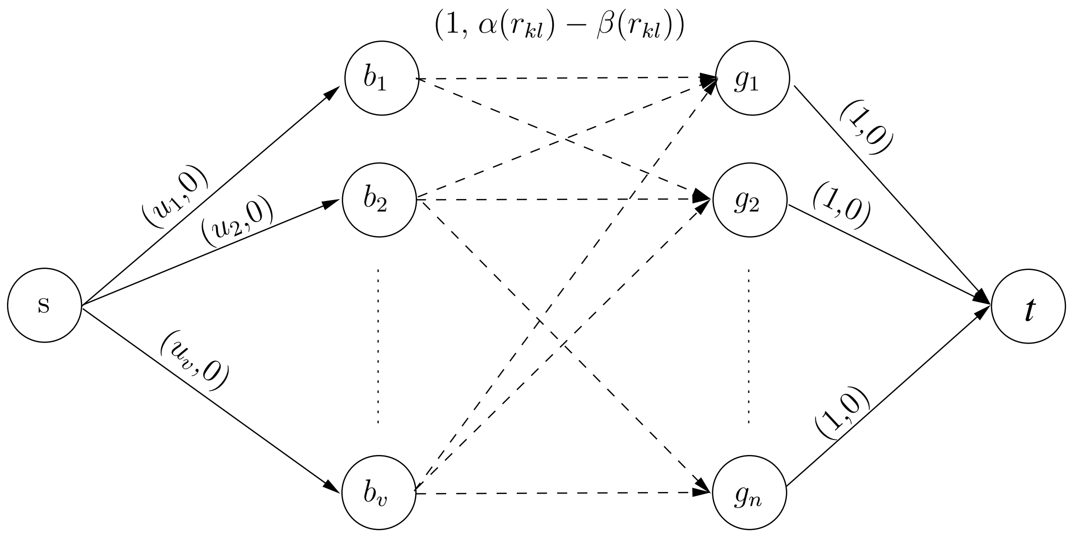

Figure 2: The network of the budget-relaxed WDP with (capacity,cost) values on the arcs.

This study introduces three problem specific heuristic methods in the order of increasing complexity for the WDP. In each of these methods, firstly, all the bids in a given DABC problem instance P are partitioned into subbids and a list L of all subbids in P is generated. For instance, if three goods are requested in a bid, i.e. $R_k = \{Book\ A, Book\ B, Book\ C\}$, a subbid is generated for each of the books A, B and C. A *subbid* is a data structure which comprises, at least, the good requested inside the subbid, the associated bid price, the bid number to which the subbid belongs, and a flag indicating whether the subbid is satisfied or not.

After the list L is generated, it is sorted according to a given sorting criterion C. In this study, the following three different values for C are used:

(i) *Unsorted*: L is kept as it is, i.e. in the order of arrival time,

(ii) *Utility*: L is sorted in descending order by the utility values of the subbids, and

(iii) *Linear Relaxation Values*: L is sorted in descending order by the value of the decision variables x_{kl} obtained when the linear relaxation of the WDP is solved.

Note that sorting criterion *Unsorted* is included for demonstrating the effects of the other two sorting criteria on the performances of heuristic methods.

The first heuristic method is called *SatisfySubbids* (SS) which can be seen in Alg. 1. First, the list L is generated and sorted as explained above. After the list S is sorted, the first subbid in the list L (marked as the current subbid) is checked whether it can be satisfied or not. A subbid is satisfiable if:

(i) the subbid is unsatisfied,

(ii) the good requested in the subbid is available, i.e. not sold,

(iii) the number of other subbids that are already satisfied in the same bid is less than the upper purchase limit value of that bid, and finally,

(iv) the current balance of the buyer is greater than or equal to the price of the subbid.

If the current subbid is satisfiable then it is committed, meaning that the subbid is marked as satisfied, the requested good is marked as sold, and the price of the subbid is subtracted from the balance of the buyer and added to the balance of the seller. Initially, the balance of each bidder is set to the spending limit value declared by that bidder. The algorithm then moves to the next subbid.

Algorithm 1 SatisfySubbids

Input: DABC problem instance P, sorting criterion C for subbids.
Output: Solution list S of satisfiable subbids.

Set $S \leftarrow \{\}$
Build a list L of all subbids in P and mark all subbids in L as *unsatisfied*
Sort L according to sorting criterion C
Set $currIndex \leftarrow 0$ // Indices start from 0
while $currIndex < |L|$ **do**
    Set $currSubbid \leftarrow L[currIndex]$
    Set $nextIndex \leftarrow currIndex + 1$
    if $currSubbid$ is satisfiable *with* balance check **then**
        Commit $currSubbid$
        Add $currSubbid$ to S
        // Find the next subbid to be checked
        for all $subbid' \in L$ of the bidder who owns of the good requested in $currSubbid$ **do**
            Set $nextIndex' \leftarrow$ the index of $subbid'$
            if $nextIndex' < nextIndex$ **and** $subbid'$ is satisfiable *with* balance check **then**
                Set $nextIndex \leftarrow nextIndex'$
            end if
        end for
    end if
    Set $currIndex \leftarrow nextIndex$
end while
return

The next subbid is determined as follows. If the current submit is not committed, i.e. unsatisfiable, then the current subbid is skipped and the next subbid is simply the subbid that immediately follows the current subbid in the list L. However, if the current submit is committed, firstly, the minimum index of the satisfiable subbids of seller is found and compared to the index of the current subbid. If the former is smaller, then the next subbid to be checked is the former subbid. The reason for this approach is that a previously checked unsatisfiable subbid, can now be satisfiable if the buyer has now enough budget. The algorithm terminates when there is no satisfiable subbid left in L.

In the SS method, a subbid is committed if the buyer has enough budget to purchase the good requested at the time the subbid is checked. If not, that subbid is skipped. However, it may be the case that the bidder may have enough budget if she sells her goods first. In order to utilize this idea, a second heuristic method, called *SatisfySubbidsandIncreaseBudget* (SSIB), is proposed the pseudocode of which can be seen in Alg. 2.

In the SSIB method, when a subbid is checked, if the buyer has not enough budget to purchase the good, then the method tries to improve the income of the buyer. For this reason, in the SSIB method, satisfiability definition of a subbid is temporarily relaxed such that a subbid is considered as satisfiable is even if the buyer has not enough budget, i.e. condition (iv) above is not checked. If the current subbid is satisfiable without balance check, then it is temporarily committed which may or may not cause a deficit for the buyer. If the buyer is deficit, then the method tries to commit the unsatisfied *incoming* subbids to the buyer, i.e. the subbids in which goods offered by the buyer are requested, until the budget deficit is closed. The order at which the incoming subbids are processed is the same as the list L. These steps are demonstrated in Alg. 3. If by this process the budget deficit of the buyer is not closed, then the current subbid and all the incoming subbids that are already committed are rollbacked. The method continues with the next subbid.

The algorithm for determining the next subbid in the SSIB method is different than that of the SS method. The reason is that if at least one incoming subbid is committed then not only the budget of the seller of the good in the current subbid but also the budget of the buyer of the current subbid may be increased. Therefore, the subbids of both the buyer and the seller should be checked and the smallest index of the satisfiable (without balance check) subbids of these bidders should be determined. If this index value is smaller than the index of the current subbid in the list L, then the algorithm jumps to this subbid. Otherwise, the next subbid is simply the subbid which immediately follows the current subbid in the list L. The algorithm

Algorithm 2 SatisfySubbidsandIncreaseBudget

Input: DABC problem instance P, sorting criterion C for subbids.

Output: Solution list S of satisfiable subbids.

```
Set S ← {}
Build a list L of all subbids in P and mark all subbids in L as unsatisfied
Sort L according to sorting criterion C
Set currIndex ← 0 // Indices start from 0
while currIndex < |L| do
  Set currSubbid ← L[currIndex]
  Set nextIndex ← currIndex + 1
  if currSubbid is satisfiable without balance check then
    Commit currSubbid
    Set deficitClosed ← true
    currBidder ← the bidder who submitted currSubbid
    if the budget of currBidder is deficit then
      Set [deficitClosed, L_IBICommit] ← IncreaseBudgetIn(P, currBidder)
    end if
    if deficitClosed then
      Add currSubbid and L_IBICommit to S
      // Find the next subbid to be checked
      for all subbid′ ∈ L of currBidder and of the bidder who owns the good requested in
      currSubbid do
        Set nextIndex′ ← the index of subbid′
        if nextIndex′ < nextIndex and subbid′ is satisfiable without balance check then
          Set nextIndex ← nextIndex′
        end if
      end for
    else
      Rollback currSubbid
    end if
  end if
  Set currIndex ← nextIndex
end while
return
```

Algorithm 3 IncreaseBudgetIn

Input: DABC problem instance P, the bidder *currBidder* whose budget to be increased

Output: Boolean *deficitClosed* indicating whether the budget deficit of *currBidder* is closed, the list of subbids $L_{IBICommit}$ committed for increasing the budget of *currBidder*.

Set $L_{IBICommit} \leftarrow \{\}$
Set $deficitClosed \leftarrow$ **false**
for all $subbid' \in L$ in which a good owned by *currBidder* is requested **do**
    if $subbid'$ is satisfiable *with* balance check **then**
        Commit $subbid'$
        Add $subbid'$ to $L_{IBICommit}$
        if the budget deficit of *currBidder* is closed **then**
            Set $deficitClosed \leftarrow true$
            return
        end if
    end if
end for
Rollback all the subbids in $L_{IBICommit}$
return

terminates when there is no satisfiable subbid left in L.

The third proposed method, *SatisfySubbidsandImprove* (SSIMP), puts more effort into satisfying the current subbid compared to the SSIB method. In the previous two methods, when a subbid is committed, it is never rollbacked and always included in the final solution (except the temporarily committed subbids in the SSIB method without available budget). However, in the SSIMP method, a previously committed, i.e. satisfied, subbid now can be rollbacked so as to increase the total utility. The pseudocode for the SSIMP method can be seen in Alg. 4.

In this method, if the current subbid is not satisfiable because of the conditions (ii), (iii), and/or (iv) listed above, the method tries to fix these violations by committing and rollbacking a set of subbids. The summary of this method is as follows:

(i) If the good requested in the current subbid is already sold and the utility of the current subbid is greater than the utility of the subbid in which that good is purchased, then the later subbid is rollbacked, making the good available again. If by this process, the budget of the previous buyer becomes deficit, the method tries to close her budget by committing the incoming subbids to this buyer as in the SSIB method (see Alg. 3). If the budget deficit of the previous buyer cannot be closed, the current subbid is skipped and all the temporarily committed subbids are rollbacked. These steps are presented in Alg. 5.

(ii) If the number of goods purchased in the bid to which the subbid belongs is the same as the upper purchase limit value of that bid, then the already satisfied subbid inside the same bid that has the highest index in L is found. If the utility of this already satisfied subbid is less than the utility of the current bid, then this already satisfied subbid is rollbacked, enabling one more good to be purchased in this bid. If not, the current subbid is skipped. These steps are presented in Alg. 6.

(iii) Finally, if the current balance of the buyer is less than the price of the current subbid, then the method tries to increase the balance of the buyer by committing unsatisfied *incoming* subbids to the buyer as in the SSIB method (see Alg. 3). However, if the budget of the buyer cannot be increased sufficiently by applying these steps, then, different from the SSIB method, the SSIMP method rollbacks already committed *outgoing* subbids of the buyer, i.e. the subbids submitted by the buyer, one by one until the balance is sufficient for the current subbid. The order in which the outgoing subbids are processed is the reverse of the order of L. Of course, an outgoing subbid can only be rollbacked if the owner of the good requested in the subbid would not have a budget deficit. If after rollbacking the outgoing subbids, the balance of the buyer is still not sufficient, then all the temporary changes are undone and the current subbid is skipped. These steps are demonstrated in Alg. 7.

If after these three steps the current subbid becomes satisfiable and the utility to be gained from the current subbid is higher than the utility to be lost by the changes done in these three steps, then it is committed. If the current subbid is committed, then the method starts traversing the list L from the beginning, since balances of multiple bidders may be affected.

Note that, none of these methods show a cyclic behavior, since an action which would cause a decrease in the total utility is not accepted, i.e. the total utility does not decrease throughout the iterations.

6 Experimental Results

In order to estimate the performances of the proposed problem specific heuristic methods under real-life market conditions, a test case generator was developed.

The test case generator uses GNU Scientific Library. It has ten parameters which can be configured to design a factorial testing. These parameters are as follows:

Algorithm 4 SatisfySubbidsandImprove

Input: DABC problem instance P, sorting criterion C for subbids.

Output: Solution list S of satisfiable subbids.

```
Set S ← {}
Build a list L of all subbids in P and mark all subbids in L as unsatisfied
Sort L according to sorting criterion C
Set currIndex ← 0 // Indices start from 0
while currIndex < |L| do
  Set currSubbid ← L[currIndex]
  Set nextIndex ← currIndex + 1
  if currSubbid is not satisfied then
    Set currGood ← the good requested in currSubbid
    if currGood is already sold then
      Set [goodRecovered, L_RGCommit, s_RGRollback] ← RecoverGood(P, currGood)
      if not goodRecovered then
        Set currIndex ← nextIndex and continue
      end if
    else
      Commit currSubbid
    end if
    Set currBid ← the bid containing currSubbid
    if the upper purchase limit value of currBid is negative then
      Set [uLimitFixed, s_FULRollback] ← FixULimit(P, currBid)
      if not uLimitFixed then
        Undo all the changes made in this iteration
        Set currIndex ← nextIndex and continue
      end if
    end if
    currBidder ← the bidder who submitted currSubbid
    if the budget of currBidder is deficit then
      Set [deficitClosed, L_IBICommit] ← IncreaseBudgetIn(P, currBidder)
      if not deficitClosed then
        Set [deficitClosed, L_IBORollback] ← IncreaseBudgetOut(P, currBidder)
      end if
      if not deficitClosed then
        Undo all the changes made in this iteration
        Set currIndex ← nextIndex and continue
      end if
    end if
    Add currSubbid, L_RGCommit, L_IBICommit to S and Remove s_RGRollback, s_FULRollback,
    L_IBORollback from S (Discard uninitialized variables if any).
    Set nextIndex ← 0
  end if
  Set currIndex ← nextIndex
end while
return
```

Algorithm 5 RecoverGood

Input: DABC problem instance P, the good *currGood* to be recovered which is already sold.

Output: Boolean *goodRecovered* indicating whether the *currGood* is recovered or not, the list of subbids $L_{RGCommit}$ committed for increasing the budget of the previous owner of the recovered good, the rollbacked subbid $s_{RGRollback}$ in which the *currGood* is previously purchased.

Set $s_{RGRollback} \leftarrow$ the subbid in which *currGood* is already purchased
Set *prevOwner* $\leftarrow$ the bidder who submitted $s_{RGRollback}$
Set *goodRecovered* $\leftarrow$ **false**
if the utility of *currSubbid* is greater than the utility of $s_{RGRollback}$ **then**
  Rollback $s_{RGRollback}$ and Commit *currSubbid*
  Set *goodRecovered* $\leftarrow$ **true**
  if the budget of *prevOwner* is deficit **then**
    Set $[deficitClosed, L_{RGCommit}] \leftarrow IncreaseBudgetIn(P, prevOwner)$
    if not $deficitClosed$ **then**
      Rollback *currSubbid* and Commit $s_{RGRollback}$
      Set *goodRecovered* $\leftarrow$ **false**
      return
    end if
  end if
end if
return

Algorithm 6 FixULimit

Input: DABC problem instance P, the bid *currBid* whose upper purchase limit value is to be fixed.

Output: Boolean *uLimitFixed* indicating whether the upper purchase limit of *currBid* is fixed or not, the rollbacked subbid $s_{FULRollback}$ for fixing the upper purchase limit of *currBid*.

Set *uLimitFixed* $\leftarrow$ **false**
Set *utilityGain* $\leftarrow$ the amount of utility gain obtained so far in this iteration
for all *subbid'* in *currBid* (traversed in the reverse order of L) **do**
  if *subbid'* is satisfied **and** the budget of the bidder who submitted *currBid* is currently greater than or equal to the price of *subbid'* **and** the utility of *subbid'* is less than *utilityGain* **then**
    Rollback *subbid'*
    Set $s_{FULRollback} \leftarrow$ *subbid'*
    Set *uLimitFixed* $\leftarrow$ **true**
    return
  end if
end for
return

Algorithm 7 IncreaseBudgetOut

Input: DABC problem instance P, the bidder $currBidder$ whose budget to be increased

Output: Boolean $deficitClosed$ indicating whether the budget deficit of $currBidder$ is closed, the list of tuples $L_{IBORollback}$ rollbacked for increasing the budget of $currBidder$.

Set $L_{IBORollback} \leftarrow \{\}$
Set $deficitClosed \leftarrow$ **false**
Set $utilityGain \leftarrow$ the amount of utility gain obtained so far in this iteration
for all $subbid' \in L$ submitted by $currBidder$ (traversed in the reverse order of L) **do**
  if $subbid'$ is satisfied **and** the budget of the owner of the good requested in $subbid'$ is currently greater than or equal to the price of $subbid'$ **and** the utility of $subbid'$ is less than $utilityGain$ **then**
    Rollback $subbid'$
    Add $subbid'$ to $L_{IBORollback}$
    if the budget deficit of $currBidder$ is closed **then**
      Set $deficitClosed \leftarrow true$
      return
    end if
  end if
end for
Commit all the subbids in $L_{IBORollback}$
return

(i) *The market type* parameter determines which types of goods are available in the market and their price ranges. The generator supports three different market types, namely *Book*, *CD/DVD* and *Electronic* markets the price profiles for which are obtained from the studies of Ghose et al. (2006), Smith and Telang (2008) and Ghose (2009), respectively. The price profiles are based on the sales data obtained from the Amazon.com marketplace.

(ii) *The number of bidders* ranges from 1000 to 20,000, simulating small to large-scale markets.

(iii) *The number of goods to be sold per bidder, the number of bids per bidder* and *the request set size* parameters are assumed to be distributed with Poisson distribution with means ranging from 1 to 5, simulating market scenarios with different asks to bids ratios.

(iv) *The request set determination method* defines how the goods to be purchased are determined inside the bids. There are two approaches implemented, either the requested goods in a bid are selected uniformly among the available goods in the market or only the first requested good is selected uniformly among the available goods and the rest are selected uniformly from a set of goods whose value are close to the value of the first good. The former approach simulates the markets in which substitutable goods have wide range of prices whereas in the latter approach, the substitutable goods assumed to be priced closely to each other.

(v) *The upper purchase limit* of a bid is uniformly selected between 1 and the size of the request set of the bid.

(vi) *The spending limit ratio* parameter is used to determine the spending limits of the bidder. The spending limit ratio takes five different values between 5% and 75% in the generated test cases. Note that when the spending limit ratio is close to 0%, the bidder can only spend the income obtained from her sold goods for purchasing new ones. On the contrary, when the spending limit ratio is close to 100%, the bidder is expected to be able to purchase the goods she wants without the need of income obtained from her sold goods.

(vii) k is the parameter of k-DA policy. $k = 0.5$ is used in the test cases.

Using the generator, a comprehensive test suite consisting of 8100 DABC model instances was prepared. These test instances were solved using a workstation with two 8-core 3.1 GHz Intel Xeon processors and a total of 128 GB memory using Linux operating system. For solving the linear integer program of the model, Gurobi Mixed Integer Programming (MIP) solver version 7 was used. The solver was given a wall-clock time limit of 60 minutes for each instance.

Among the generated 8100 problem instances, the MIP solver was able to find the optimal solutions for 6650 instances (approximately 82% of the instances) using the default optimality gap of 0.01%. The *optimality gap* of a solution is defined as the ratio of the difference between the best integer solution and the best upper bound to the best upper bound. These instances are referred as *optimally solved instances* throughout the text. For the remaining 1450 instances, the MIP solver could not find the optimal solution within the given time limit of 60 minutes, however, it was able to find an integer feasible solution for each of these test cases. These more difficult instances are, on the other hand, referred as *sub-optimally solved instances.*

Each instance was also solved by each of the proposed three heuristic methods, SS, SSIB, and SSIMP and for each method, three different sorting criteria, *unsorted*, *utility* and *linear relaxation values*, were used. Thus, each instance was solved using nine different configurations. In order to present the quality of the solutions found by the heuristic methods, a *goodness* measure is defined such as:

$$\text{Goodness of a Solution} = \frac{\text{Obj. Val. of the Solution Found by the Heuristic Method}}{\text{Obj. Val. of the Solution Found by the MIP Solver}} \cdot 100\%$$

The mean and standard deviation of goodness of the solutions found by the proposed heuristic methods for the optimally solved 6650 instances can be seen in Table 1 and the corresponding box plot is provided in Figure 3. The SSIMP heuristic provides the best results using the *linear relaxation values* as the sorting criterion with a mean goodness value greater than 96% with a standard deviation of less than 8%. The same heuristic with the *utility* sorting criterion performs slightly worse compared to the *linear relaxation values* case with 1% less mean goodness value. The results of the SSIB heuristic is behind the results of the SSIMP heuristic by a small margin having an average of approximately 94% goodness independent of whether the *utility* or the *linear relaxation values* sorting criterion is used. Finally, despite being the simplest of the proposed heuristics, the SS heuristic provides quite acceptable results having a mean goodness value of approximately 90% and 92% using the *utility* and the *linear relaxation values* sorting criteria, respectively.

Table 1: Goodness of solutions of the heuristic methods for the optimally solved 6650 problem instances (optimality gap $\leq 0.01\%$).

	SS		SSIB		SSIMP	
Sorting Criterion	mean	stdev	mean	stdev	mean	stdev
Unsorted	**54.0%**	11.8%	**55.3%**	11.6%	**92.6%**	9.7%
Utility	**89.5%**	11.0%	**93.9%**	8.8%	**95.3%**	7.7%
Lin. Rel. Values	**92.4%**	11.0%	**93.9%**	9.6%	**96.3%**	7.8%

The results for the *unsorted* sorting criterion, which can also be regarded as random order, indicate that ordering of subbids is important for all three heuristic methods, however, the SSIMP heuristic much more less affected by the sorting criterion, performing quite well even with a randomly sorted subbid list which contributes to the reliability of the results found by this heuristic method. This is not the case for the SS and SSIB methods which rely on the quality of the sorting criterion heavily such that an unsorted subbid list may cause more than 30% decrease in the mean goodness.

In order to understand whether the presented pair-wise mean differences of the solutions found by the heuristic methods are statistically significant or not, an ANOVA test is performed with an $alpha = 0.01$ significance level followed by a multi comparison analysis testing. The test results reveal that all the mean differences other than two are statistically significant at this significance level which are indicated in Figure 3.

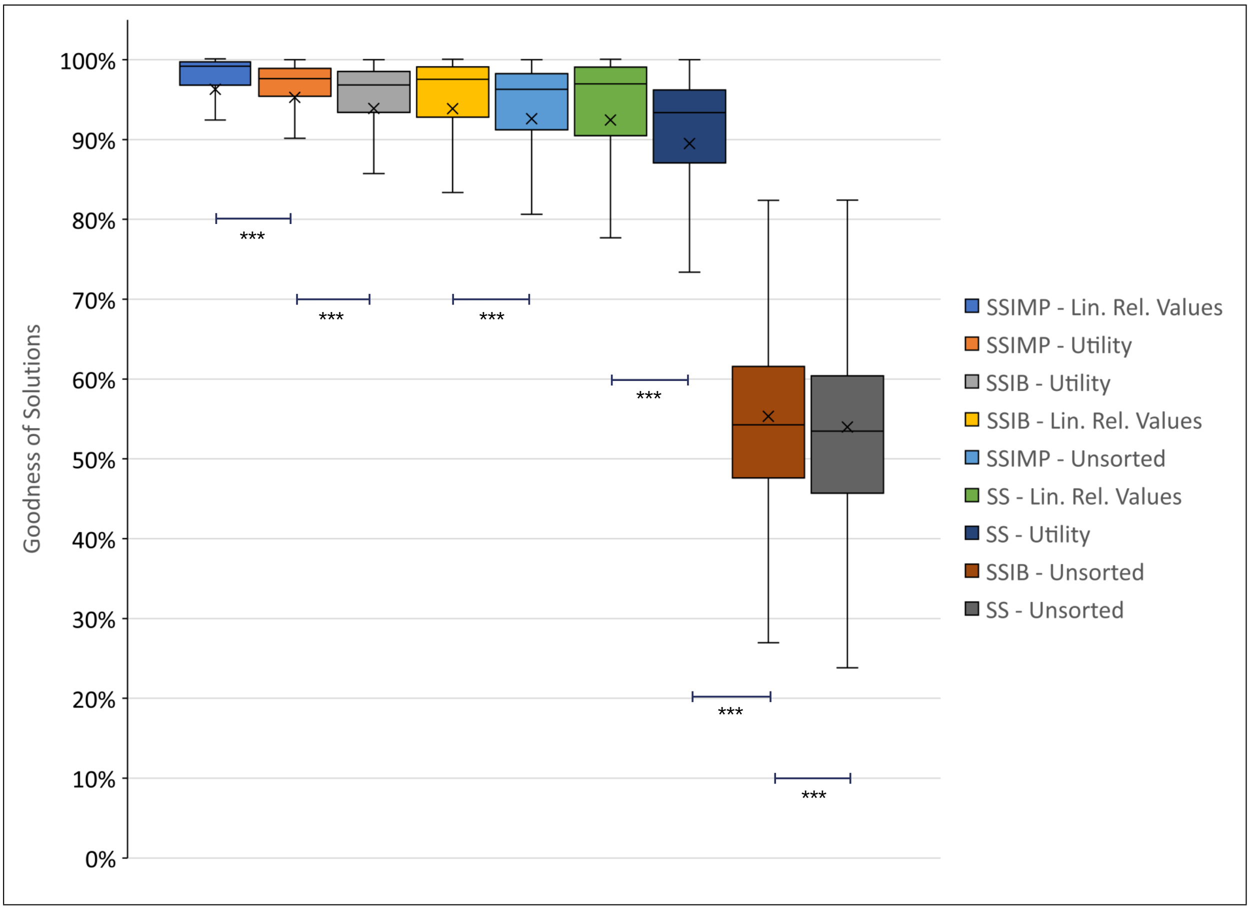

Figure 3: Boxplot presenting the goodness of solutions of the heuristic method-sorting criterion pair for the optimally solved 6650 problem instances (optimality gap $\leq 0.01\%$). An 'x' indicates the mean value of the corresponding goodness values and a horizontal bar with '***' indicates that the corresponding mean difference is statistically significant at the $\alpha = 0.01$ level.

The performances of the proposed heuristic methods are also measured for the remaining sub-optimally solved 1450 problem instances ()with an optimality gap higher than 0.01%) with respect to the sub-optimal solutions found by the MIP solver. Note that, some of these solutions could in fact be optimal, however, the MIP solver might not have proven the optimality of the solutions within the given time limit. The mean and standard deviation of goodness of the solutions found by the heuristic methods for these instances are reported in Table 2. Note that, in this case the goodness values may be higher than 100%, meaning that proposed heuristic methods may find better solutions than the MIP solver. Actually, this is the case and it is seen that SS and SSIB methods with the *utility* or linear relaxation values provide approximately 60% and 90% better solutions than the MIP solver (with the given time limit of 60 minutes for each instance) on average, respectively. For the SSIMP method, more than two-fold increase in the total utility is observed on average for these difficult cases.

Table 2: Goodness of solutions of the heuristic methods for the sub-optimally solved 1450 problem instances (optimality gap > 0.01%).

	SS		SSIB		SSIMP	
Sorting Criterion	mean	stdev	mean	stdev	mean	stdev
Unsorted	**90%**	71%	**103%**	83%	**185%**	151%
Utility	**159%**	125%	**193%**	158%	**213%**	180%
Lin. Rel. Values	**168%**	131%	**189%**	153%	**215%**	180%

The mean running times of the heuristic methods and the MIP solver for all problem instances can be seen in Table 3. The SS and SSIB heuristics are very fast when using the *utility* as the sorting criterion, finding the solution in less than 2 seconds and 1 second respectively on average even for the largest test cases with 20,000 bidders. Although still fast, when the *linear relaxation values* sorting criterion is used, the mean running time increases to approximately 44 seconds for the largest test cases and 11 seconds on average. The reason for this increase is the time required for solving the linear relaxation of the integer program introduced in Section 4 for each of the test instances.

Table 3: Mean running times of the heuristic methods and the MIP solver in seconds for all problem instances categorized into the number of bidders. Note that the maximum running time of the MIP solver for each instance is set to 3600 seconds.

Heuristic	Sorting	Number of Bidders					
Method	Criterion	1000	2000	5000	10000	20000	Overall
SS	Unsorted	0.0	0.0	0.1	0.4	1.6	**0.4**
	Utility	0.0	0.0	0.1	0.3	1.5	**0.4**
	Lin. Rel. Values	0.1	0.4	2.2	8.8	44.3	**11.2**
SSIB	Unsorted	0.0	0.0	0.0	0.1	0.3	**0.1**
	Utility	0.0	0.0	0.0	0.1	0.4	**0.1**
	Lin. Rel. Values	0.1	0.4	2.2	8.6	43.7	**11.0**
SSIMP	Unsorted	1.1	4.8	37.1	251.8	1408.9	**340.7**
	Utility	0.4	1.7	13.5	84.5	479.5	**115.9**
	Lin. Rel. Values	0.3	1.0	6.8	37.3	220.1	**53.1**
MIP Solver		456.9	549.2	711.5	842.3	985.9	**709.2**

The SSIMP heuristic takes longer to find a solution compared to other two proposed heuristics in line with its complexity. However, unlike the previous cases, the order of the subbids have a huge impact on the running times of the SSIMP heuristic. The SSIMP heuristic with the *utility* sorting criterion requires approximately 116 seconds on average whereas this reduces to approximately 53 seconds with the *linear relaxation values* sorting criterion despite the fact that the latter one requires additional time for finding the linear relaxation values as explained above. The worst running times are obtained when the subbids are not sorted at all. The reason for this difference is that the SSIMP method may work less on improving steps depending on how the subbids are sorted.

Comparison of the Proposed Problem Specific Heuristics and Standard Genetic Algorithm Based Approaches for the WDP

The presented results so far demonstrate the performances of the proposed problem specific heuristics with respect to that of the general purpose MIP solver in terms of both quality and running time. However, a question may arise whether a standard meta-heuristic approach would also provide the same results or even better results than the proposed methods. For this purpose, three Genetic Algorithm (GA) based heuristic methods were implemented. In these methods, each solution is encoded as a binary string whose length is the number of subbids where each bit in the string indicates whether the corresponding subbid is satisfied or not in the corresponding solution. However, since there may be infeasible solutions in the solution space, three different methods, namely GA-Penalty, GA-Feasible, and GA-Repair, using different approaches for handling infeasibility were implemented.

GA-Penalty method is the standard GA implementation in which solutions in the population are allowed to be infeasible. The initial population is randomly initialized. In order to solve the infeasibility problem, fitness values of infeasible solutions are penalized while evaluating the solutions. The penalty value, p, is defined as:

$$p = \max_{k,l} U_{kl} \cdot \left[\sum_{\forall k \,|\, b_k \in B} \left(\sum_{\forall l \,|\, r_{kl} \in R_k} x_{kl} - u_k \right) + \sum_{\forall j \,|\, g_j \in G} \left(\sum_{\forall k,l \,|\, b_k \in B \wedge r_{kl} \in R_k \wedge r_{kl} = g_j} x_{kl} - 1 \right) \right] + \sum_{\forall i \,|\, d_i \in D} \left(\sum_{\forall k,l \,|\, b_k \in B_i \wedge r_{kl} \in R_k} P_{kl}\, x_{kl} - \sum_{\forall k,l \,|\, b_k \in B \wedge r_{kl} \in R_k \wedge r_{kl} \in G_i} P_{kl}\, x_{kl} - l_i \right) \quad (6)$$

so that p is proportional to the amounts of violations of constraints in Eq.(2)-(4). If a constraint is not violated, the corresponding penalty term is set to zero (i.e. penalty value cannot be negative). In this standard implementation 1-point crossover operator is used with 0.9 crossover probability, and bitwise mutation operator is used with 0.01 probability.

Different from the GA-Penalty, in GA-Feasible method, infeasible solutions are not allowed in a population. Therefore, initial population is randomly generated while considering feasibility. In this method, uniform 0.5 crossover operator is used to prevent positional bias, and mutations are applied using bitwise mutation operator. However, these operators are modified such that they output only feasible solutions. This is achieved as follows: in the uniform crossover operator, corresponding bits of the two children are swapped if swapping bits does not introduce infeasibility to any of the child. Similarly, in the mutation operator bit mutation is applied if it does not cause infeasibility to the corresponding child. Since this modification to the mutation operator reduces the probability of a change in a solution, the bitwise mutation probability value is set to a higher value of 0.1 in this method.

In the third method, GA-Repair, a repair function is used for repairing infeasible solutions before evaluating their fitness values. The repair operation has two phases. In the first phase, for a given infeasible solution the subbids which cause violation in Eq.(2)-(4) are rollbacked, and in the second phase, the remaining satisfiable subbids of the bidders are committed. The order of the subbids to be rollbacked and committed is determined randomly to maintain diversity. Although the solution becomes feasible after the first phase, the second phase ensures that all solutions in a population are located at the border of the feasible region of the corresponding problem instance. The motivation is that the optimal solution is also located this border. In this method, since solutions are allowed to be infeasible until they are evaluated, standard uniform 0.5 crossover operator and bitwise mutation operator with 0.1 probability are used.

For all the GA methods, binary tournament selection method is used for the mating pool selection process, and the stopping criterion is determined to stop if no improvement in the best solution is observed in the last 100 generations.

The performances of these three methods are also measured on the same test suite as the proposed heuristics. For each method, populations sizes of 50 and 100 were used. Each method is given a wall-clock running time limit of 60 minutes for each instance. The comparison of the mean goodness of the solutions found by the GA based heuristic methods and by the proposed problem specific heuristics is given in Figure 4.

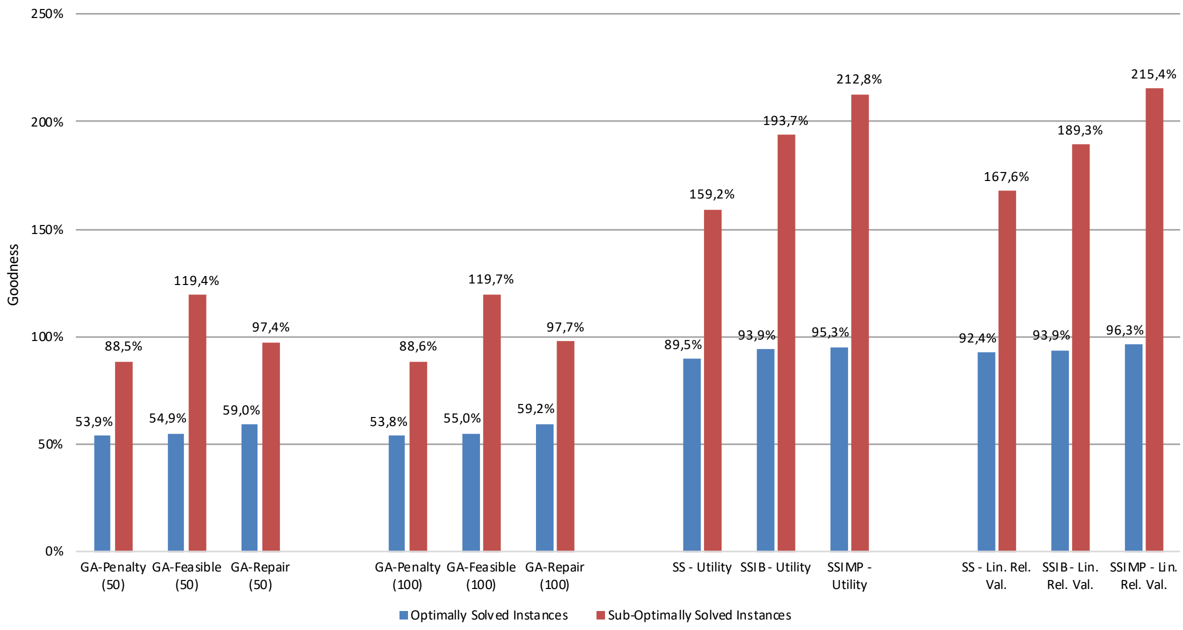

Figure 4: Mean goodness values of the GA based heuristic methods with population sizes 50 and 100, and the proposed problem specific heuristic methods.

For the optimally solved instances, mean goodness values of GA based heuristics vary approximately in between 54% and 59%, GA-Repair providing better results on average than the other GA-based methods. On the other hand, for the relatively more difficult sub-optimally solved instances, GA-Feasible method provides the best results on average among the GA-based methods with a mean goodness value of approximately 120%. Increasing population size from 50 to 100 has minimal impact, causing an insignificant increase in the mean goodness values.

The proposed heuristics, however, provide significantly better results than the GA-based methods with mean goodness values approximately in between 90% and 96% for the optimally solved instances, and in between 160% and 215% for the remaining sub-optimally solved instances.

The mean running times of the GA-based methods and the proposed heuristics for all the test instances are compared in Figure 5. GA-Repair method takes longer compared to the other two GA-based methods probably because of the expensive repair operation carried out for each infeasible solution. The impact of increasing the population size from 50 to 100 is also observed in this case, causing approximately twofold increase in the mean running times for each GA-based algorithm. It is observed that the proposed problem specific heuristics provide results faster on average than the GA-based methods with an exception for SSIMP-Utility when

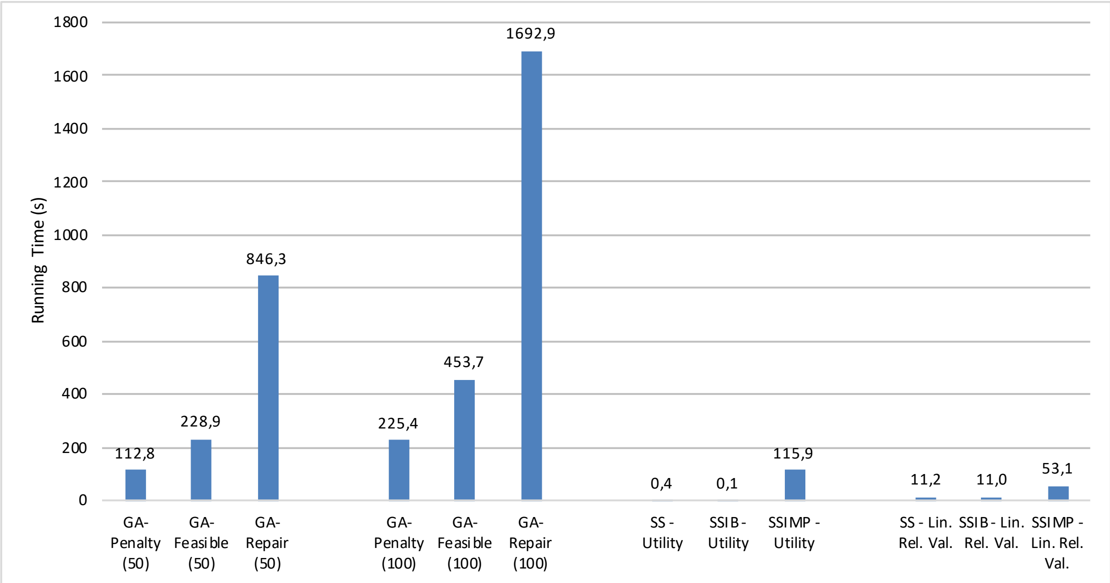

Figure 5: Mean running times of the GA based heuristic methods with population sizes 50 and 100, and the proposed problem specific heuristic methods for all problem instances.

compared to GA-Penalty method using a population size of 50.

The Total Utility Gain Provided by the DABC Model

The features of the DABC model were explained in the previous sections in detail. By means of these features, the DABC model aims to increase the economic welfare of the participants. In order to *estimate* possible welfare increase with respect to the currently used traditional market mechanism, a simulation experiment was conducted. In this experiment, each optimally solved test instance was simulated under the following traditional market rules:

- Initially each buyer has a budget equal to her spending limit.
- A buyer can purchase a good she is interested only if the good was not sold and the buyer has enough budget. If this is the case, a bidder may initiate a transaction whenever she wants. Transactions are executed in a first-come-first-served manner.
- When a buyer purchases a good, an amount equal to its price is transferred from the buyer to the corresponding seller.
- The simulation ends when no buyer can purchase any more goods they want.

For each test instance, the simulation was conducted 100 times with a different random order of purchase requests and the mean total utility gain was calculated. The utility of a

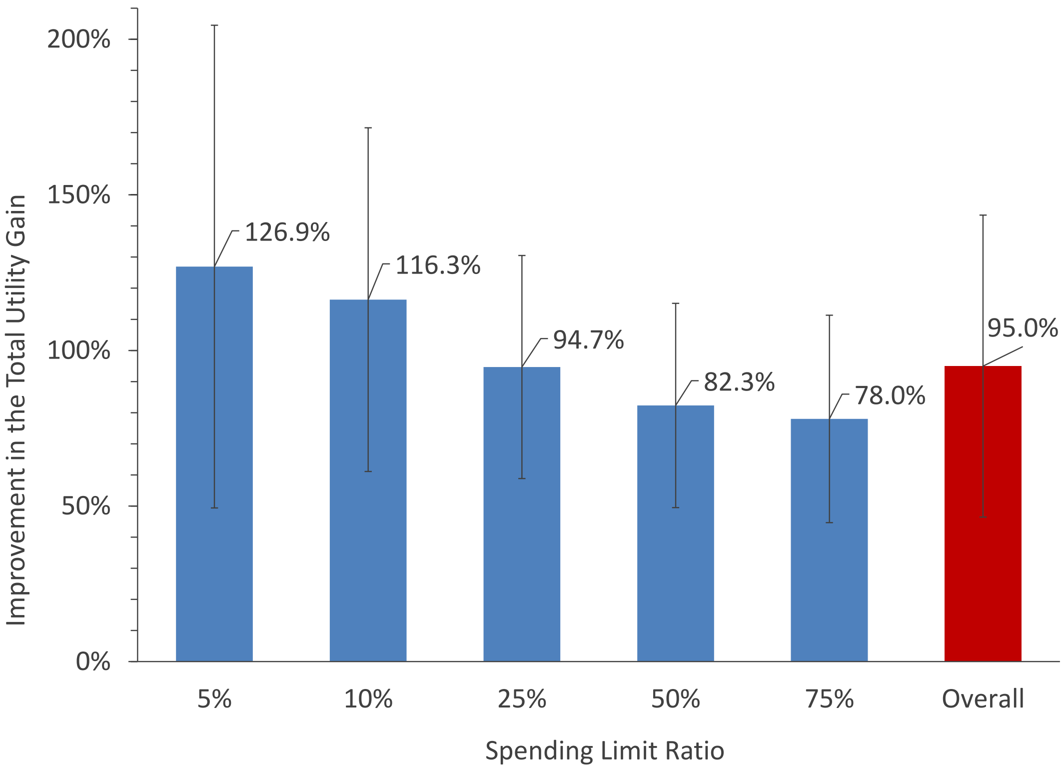

Figure 6: Mean improvement in the total utility gain using the DABC model over the first-come-first-served approach-based market simulation for the optimally solved test cases. Horizontal bars indicate the standard error of the corresponding sample.

transaction was assumed to be the same as in the DABC model. The total utility values for the test instances were compared to the total utility values obtained using the DABC model.

The results of this experiment which are grouped into the spending limit ratios can be seen in Figure 6. When the participants have initially small budgets, i.e. when the spending limit ratio is small, the mean improvement in the total utility is observed as much as approximately 127%, meaning that the total utility obtained using the DABC model is approximately 2.27 times of the simulation.

Although the improvement rate reduces as the spending limit ratio increases, the improvement in the case of 75% spending limit ratio is also quite high with a value higher than 75%. Overall mean improvement for all optimally solved test instances is 95%. That is, almost two-fold of improvement in the total utility on average is observed when the DABC model is used for these test instances using the above-mentioned simulation rules.

7 Discussion and Conclusion

In addition to providing strong environmental benefits, development of secondhand markets also increases overall economic welfare regardless of whether they shift the demand for new goods (Thomas, 2003). Although many Internet-based secondhand marketplaces have emerged in the last two decades increasing the trading volume of used goods, as also stated by Clausen et al. (2010), more potential is yet to be captured. Accordingly, this paper introduces the DABC electronic market model which aims to attract more participants to the secondhand markets and to increase their economic welfare.

The winner determination problem of the DABC model is proven to be NP-hard, and therefore, three problem specific heuristic methods are proposed each of which can be used with different sorting criteria. Based on an experiment conducted on a large test suite, the SSIB heuristic with the *utility* sorting criterion provides results within only approximately 6% of the optimum in less than half a second on average even for the largest test cases including 20,000 bidders. Thus, the SSIB heuristic is considered to be highly scalable which can be used in markets with hundreds of thousands of participants. The more sophisticated heuristic method, the SSI method using the *linear relaxation values* sorting criterion, on the other hand improves the quality of the solutions by approximately 2.5% on average providing results within less than 4% of the optimum. The running time of this heuristic, however, also increases to approximately 220 seconds for large test cases and to 53 seconds on average.

This study also presents another experiment for assessing *possible* improvement in the total utility to be obtained when the DABC institution is used instead of the current traditional market institution. The results indicate an improvement rate of approximately 95% in the total utility on average for the problem instances in the test suite, where the lowest mean improvement rate being 78 % if the bidders have more budget initially and being approximately 127 % if their initial budgets are low. This improvement is considered to occur due to the following features of the model: Firstly, in the DABC model, asks and bids are collected during a trading round and then the global optimal trading pattern is found, i.e. a batch processing is done, instead of on-line processing in the first-come-first-served based traditional markets. Secondly, by enabling the income to be obtained from the sold goods to be used for purchases, the model can extract feasible trading patterns which are otherwise not possible.

As a conclusion, it is considered that the demonstrated features of the DABC model would be useful increasing the potential of the used good trading and by means of the proposed efficient heuristic methods, the DABC model can be used in very large-scale Internet-based electronic markets with tens or even hundreds of thousands of participants. However, as a future work a more detailed economic analysis of the model would provide more insight into the efficiency of the DABC model when used in real-world markets.

Acknowledgments

This work was supported by Research Fund of the Marmara University (Project Number: FEN-A-130612-0218). I would like to thank the reviewers for their helpful comments that have helped to improve the paper.

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Prentice-Hall, Inc.

Arunkundram, R. and Sundararajan, A. (1998). An economic analysis of electronic secondary markets: Installed base, technology, durability and firm profitability. *Decision Support Systems*, 24(1):3–16. `https://dx.doi.org/10.1016/S0167-9236(98)00059-1`.

Babaioff, M. and Nisan, N. (2004). Concurrent auctions across the supply chain. *Journal of Artificial Intelligence Research*, 21:595–629. `https://dx.doi.org/10.1613/jair.1316`.

Bao, S. and Wurman, P. R. (2003). A comparison of two algorithms for multi-unit k-double auctions. In *Proceedings of the 5th international conference on Electronic commerce*, pages 47–52. ACM, Pittsburgh, Pennsylvania, USA. `https://dx.doi.org/10.1145/948005.948012`.

Chatterjee, K. and Samuelson, W. (1983). Bargaining under incomplete information. *Operations Research*, 31(5):835–851. `https://dx.doi.org/10.1287/opre.31.5.835`.

Clausen, J., Blättel-Mink, B., Erdmann, L., and Henseling, C. (2010). Contribution of online trading of used goods to resource efficiency: An empirical study of ebay users. *Sustainability*, 2(6):1810–1830. `https://dx.doi.org/10.3390/su2061810`.

Cooper, D. R. and Gutowski, T. G. (2017). The environmental impacts of reuse: A review. *Journal of Industrial Ecology*, 21(1):38–56. https://dx.doi.org/10.1111/jiec.12388.

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

Fan, M., Stallaert, J., and Whinston, A. B. (1999). A web-based financial trading system. *Computer*, 32(4):64–70. https://dx.doi.org/10.1109/2.755007.

Friedman, D. (1984). On the efficiency of experimental double auction markets. *The American Economic Review*, 74(1):60–72.

Friedman, D. (1993). The double auction market institution: A survey. In Friedman, D. and Rust, J., editors, *The Double Auction Market*, Proceedings volume in the Santa Fe Institute studies in the sciences of complexity, pages 3–25. Avalon Publishing.

Friedman, D. and Rust, J., editors (1993). *The Double Auction Market: Institutions, Theories, and Evidence*, volume 14 of *Proceedings volume in the Santa Fe Institute studies in the sciences of complexity*. Avalon Publishing.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman. https://books.google.com.tr/books?id=fjxGAQAAIAAJ.

Geissdoerfer, M., Savaget, P., Bocken, N. M., and Hultink, E. J. (2017). The circular economy – a new sustainability paradigm? *Journal of Cleaner Production*, 143(Supplement C):757–768. https://dx.doi.org/10.1016/j.jclepro.2016.12.048.

Ghisellini, P., Cialani, C., and Ulgiati, S. (2016). A review on circular economy: The expected transition to a balanced interplay of environmental and economic systems. *Journal of Cleaner Production*, 114(Supplement C):11–32. https://dx.doi.org/10.1016/j.jclepro.2015.09.007.

Ghose, A. (2009). Internet exchanges for used goods: An empirical analysis of trade patterns and adverse selection. *MIS Quarterly*, 33(2):263–291.

Ghose, A., Smith, M. D., and Telang, R. (2006). Internet exchanges for used books: An empirical analysis of product cannibalization and welfare impact. *Information Systems Research*, 17(1):3–19. https://dx.doi.org/10.1287/isre.1050.0072.

Gode, D. K. and Sunder, S. (1993). Lower bounds for efficiency of surplus extraction in double auctions. In Friedman, D. and Rust, J., editors, *The Double Auction Market*, Proceedings volume in the Santa Fe Institute studies in the sciences of complexity, pages 199–214. Avalon Publishing.

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

Huang, P., Scheller-Wolf, A., and Sycara, K. (2002). Design of a multi–unit double auction e–market. *Computational Intelligence*, 18(4):596–617. https://dx.doi.org/10.1111/1467-8640.t01-1-00206.

Kalagnanam, J. R., Davenport, A. J., and Lee, H. S. (2001). Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Research*, 1(3):221–238. https://dx.doi.org/10.1023/A:1011589804040.

Parsons, S., Marcinkiewicz, M., Niu, J., and Phelps, S. (2006). Everything you wanted to know about double auctions but were afraid to (bid or) ask: Draft. http://www.sci.brooklyn.cuny.edu/~parsons/projects/mech-design/publications/cda.pdf (accessed on Mar 2018).

Pearce, D. W. and Turner, R. K. (1989). *Economics of natural resources and the environment*. The Johns Hopkins University Press, Baltimore.

Phelps, S., Parsons, S., and McBurney, P. (2005). An evolutionary game-theoretic comparison of two double-auction market designs. In Faratin, P. and Rodríguez-Aguilar, J. A., editors, *Agent-Mediated Electronic Commerce VI. Theories for and Engineering of Distributed Mechanisms and Systems: AAMAS 2004 Workshop, AMEC 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, pages 101–114. Springer Berlin Heidelberg, Berlin, Heidelberg. https://dx.doi.org/10.1007/11575726_8.

Plott, C. R. (1982). Industrial organization theory and experimental economics. *Journal of Economic Literature*, 20(4):1485–1527.

Plott, C. R. and Gray, P. (1990). The multiple unit double auction. *Journal of Economic Behavior & Organization*, 13(2):245–258. https://dx.doi.org/10.1016/0167-2681(90)90089-V.

Rustichini, A., Satterthwaite, M. A., and Williams, S. R. (1994). Convergence to efficiency in a simple market with incomplete information. *Econometrica*, 62(5):1041–1063. https://dx.doi.org/10.2307/2951506.

Satterthwaite, M. A. and Williams, S. R. (1989). Bilateral trade with the sealed bid k-double auction: Existence and efficiency. *Journal of Economic Theory*, 48(1):107–133. https://dx.doi.org/10.1016/0022-0531(89)90121-X.

Satterthwaite, M. A. and Williams, S. R. (1993). The bayesian theory of the k-double auction. In Friedman, D. and Rust, J., editors, *The Double Auction Market*, Proceedings volume in the Santa Fe Institute studies in the sciences of complexity, pages 99–123. Avalon Publishing.

Satterthwaite, M. A. and Williams, S. R. (2002). The optimality of a simple market mechanism. *Econometrica*, 70(5):1841–1863.

Smith, M. D. and Telang, R. (2008). Internet exchanges for used digital goods. https://ssrn.com/abstract=1086358.

Smith, V. L. (1962). An experimental study of competitive market behavior. *Journal of Political Economy*, 70(2):111–137. http://www.jstor.org/stable/1861810.

Smith, V. L. (1982). Microeconomic systems as an experimental science. *The American Economic Review*, 72(5):923–955.

Stahel, W. R. (2016). The circular economy. *Nature News*, 531(7595):435. https://dx.doi.org/10.1038/531435a.

The European Commission (2015). Towards a circular economy. https://ec.europa.eu/commission/priorities/jobs-growth-and-investment/towards-circular-economy_en (accessed on Mar 2018).

Thomas, V. M. (2003). Demand and dematerialization impacts of second-hand markets. *Journal of Industrial Ecology*, 7(2):65–78. https://dx.doi.org/10.1162/108819803322564352.

Thomas, V. M. (2011). The environmental potential of reuse: An application to used books. *Sustainability Science*, 6(1):109–116. https://dx.doi.org/10.1007/s11625-010-0115-z.

United States Environmental Protection Agency (2017). Reduce, reuse, recycle. https://www.epa.gov/recycle (accessed on Mar 2018).

Wilson, R. (1985). Incentive efficiency of double auctions. *Econometrica*, 53(5):1101–1115. https://dx.doi.org/10.2307/1911013.

Wurman, P. R., Walsh, W. E., and Wellman, M. P. (1998). Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24(1):17–27. https://dx.doi.org/10.1016/S0167-9236(98)00060-8.

Xia, M., Stallaert, J., and Whinston, A. B. (2005). Solving the combinatorial double auction problem. *European Journal of Operational Research*, 164(1):239–251. https://dx.doi.org/10.1016/j.ejor.2003.11.018.

Yuan, Z., Bi, J., and Moriguichi, Y. (2006). The circular economy: A new development strategy in china. *Journal of Industrial Ecology*, 10(1-2):4–8. https://dx.doi.org/10.1162/108819806775545321.