Proceedings of the Federated Conference on Computer Science and Information Systems pp. 459–467
DOI: 10.15439/2017F291
ISSN 2300-5963 ACSIS, Vol. 11

An Electronic Market Model with Mathematical Formulation and Heuristics for Large-Scale Book Trading

Ali Haydar Özer
Department of Computer Engineering,
Marmara University,
34722, Istanbul, Turkey.
Email: haydar.ozer@marmara.edu.tr

***Abstract*—This study introduces an electronic market model for secondary book markets in which each market participant can put up books for sale, and simultaneously place requests for book purchase. The model allows participants to declare a budget limit so that for each participant, the difference between the cost of purchased books and the revenue obtained from sold books stays within the declared budget limit. The model also allows participants to declare sets of substitutable books along with their preferences so that they can purchase at most one book from each of these sets. In this study, the mathematical definition of the market model is introduced, and the corresponding winner determination problem is formulated as a multi-objective linear integer program. Since this problem is NP-Hard, three heuristic methods are proposed and the performances of these methods are demonstrated on a comprehensive test suite. The results indicate that the model can be used efficiently in large-scale electronic markets in which durable goods are exchanged with tens of thousands of participants.**

I. Introduction

RECENT advances in information technology provided a shift from traditional physical markets where the participants meet at a certain place for exchanging commodities to the electronic markets. Electronic markets provide a platform bringing multiple buyers and sellers in contact by weakening space and time restrictions [1]. Therefore, an electronic market has the potential of attracting more participants than a physical market. For instance, eBay, the world's largest online market, has more than 160 million buyers globally [2] and Alibaba.com, the world's biggest business-to-business market has more than 400 million active buyes [3]. As the number of participants increases, the higher competition level among the suppliers causes increased supplier innovation [4]. An e-market can reduce buyers' search costs to obtain information about the product offerings of sellers [5], [6]. This increases the allocative efficiency of the market, i.e. the efficiency with which a market is allocating resources [7].

This study focuses on secondary electronic book markets, that is electronic markets for both used and new book trading. Secondary book markets play an important role in overall economic activity with multi-billion dollars of transaction volumes. For instance, in the U.S., the transaction volume of the used book market was approximately $2.2 billion in 2004, and online booksellers are responsible for two-thirds of the general interest used book sales [8]. Also, compared to physical markets, electronic book markets provide increased book variety. For instance, according to the study of Brynjolfsson et al. [9], amazon.com and barnesandnoble.com have 2.3 million books listed on their online markets whereas a typical brick-and-mortar bookstore has 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 books in their online version, walmart.com.

In this paper, an electronic market model, called *EMBook model*, is proposed which is designed especially for secondary book markets for the trading of used books as well as new ones. In this model, market participants can have both buyer and seller roles, meaning that each participant can simultaneously put forward books for sale as well as declare requests for purchase. Thus, the market allows participants to spend the revenue to be obtained from the books they want to sell for the books they want to buy. The model also offers a budget limiting mechanism such that for each participant the amount spent on purchased books minus the revenue to be obtained from sold books does not exceed the declared budget limit of the participant. Thus, this model enables participants with limited budgets to purchase new books using the revenue from their books to be sold and also encourages them to participate in the market without a risk of having a budget deficit. Additionally, a participant may also be indifferent to multiple books, for instance there may be multiple sellers of the same book or the participant may be interested in a specific set of novels in a book market. The EMBook model further provides a mechanism for handling such situations so that in her purchase request, a participant can declare a list of substitutable books among which she wants to purchase only one. Furthermore, she is also allowed to indicate her preferences for the books she wants to purchase. By means of these features, the EMBook model aims *to attract more participants* to the used book

This work is supported by Marmara University, Scientific Research Projects Committee (BAPKO) under D-Type Project.

markets and *to increase the allocative efficiency* of such markets.

In the next section, the EMBook model is explained in detail on an example book market scenario. In Section III, the mathematical definition of the EMBook model is given, and the corresponding winner determination problem is formulated using multi-objective linear integer programming. The complexity results are also presented. Since the winner determination problem is NP-Hard, three heuristic methods are designed which are introduced in Section IV. The experimental results demonstrating the performances of the heuristic methods on a comprehensive test suite are presented in Section V. Finally, the paper is concluded in Section VI.

II. The EMBook Model

In this section, the EMBook market model and its rules are going to be introduced. In the EMBook model, each participant may sell and purchase books simultaneously, that is each participant may have a seller role, a buyer role or both. First of all, the participants with a seller role declare the books they want to sell along with the prices they request which are called *sales requests*. Thus, in this model, each book to be sold is considered as a unique item and its price is determined by its owner. This feature allows buyers to differentiate between the copies of the same book sold by different sellers, since, for instance, condition of the book, reputation of its seller, location of the seller and the associated transfer cost may vary.

Secondly, the participants with a buyer role declare the books they want to purchase which are called *purchase requests*. However, there may be multiple instances of the same book in the market (e.g. multiple copies sold by possibly different participants), or a participant may be indifferent to a number of different books (e.g. a set of novels). Considering these cases, the participants are allowed to declare one or more sets of books (called *request sets*) among which the participant can buy only one book. Each request set constructed by a participant indicates that the participant is interested in any book in this set, however, she is willing to buy only one of the books inside this set. Furthermore, if the participant is not totally indifferent to the books in the request set she declared, the request set may also be defined as an ordered set indicating the relative preferences of the participant for the books inside this set. That is, if the request set of a participant contains $\{Book\ A, Book\ C, Book\ B\}$ in this particular order, the participant is assumed to prefer $Book\ A$ over $Book\ C$, and $Book\ C$ over $Book\ B$. Note that although the participant is limited to purchase only one book, this is not a limitation for a participant who wants to purchase more, since the model also allow submission of the same request more than once, that is the purchase requests are not needed to be unique in this model.

Thirdly, after the sale and purchase requests are collected, each participant with a buyer role declares a budget limit. The budget limit indicates the maximum amount of money that the participant is willing to spend in the market. If the participant has also a seller role, the budget limit indicates the maximum difference between the expenditure and the revenue. In other words, for each participant the amount spent on the purchased books minus the revenue obtained from the sold books cannot exceed the budget limit of the corresponding participant.

In order to make the market process easier to understand, an example scenario which is illustrated in Figure 1 is provided. In this scenario, there are four participants who put up six books ($BookA$ to $BookF$) for sale with prices ranging from €15 to €40. For instance, $Participant\ 1$ wants to sell two books, $Book\ A$ and $Book\ B$, for €30 and €20, respectively. Additionally, she wants to purchase either $Book\ C$ or $Book\ D$ indicating that she prefers $Book\ C$ over $Book\ D$. For all possible outcomes, she declares that she is willing to spend at most €10. Since the price of each of $Book\ C$ and $Book\ D$ exceeds the budget limit of $Participant\ 1$, this participant cannot purchase any of these two books unless at least one of her books is sold in the market. Similarly, $Participant\ 2$, wants to sell two books, $Book\ C$ and $Book\ D$. However, this participant has two purchase requests. She wants to purchase *both* $Book\ A$ *and* one of the books from the set containing $Book\ E$, $Book\ F$ and $Book\ B$. She also declares that she prefers $Book\ E$ over $Book\ F$, and prefers $Book\ F$ over $Book\ B$. Declaring a budget limit of 0 implies that her two purchase requests can only be satisfied if both of her books are sold.

The primary aim of the EMBook model is to increase the allocative efficiency of the book market by allowing participants to use revenue to be obtained from sold books for purchasing new books. The benefit of this feature can also be seen in this scenario. The budget limits of the participants do not allow them to purchase the books they want. Therefore, in traditional book markets, first they would have to sell their books, and then they would be able to purchase new books using the obtained budget. Thus, in this particular scenario, no participants would be able to buy a book. However, the market outcome of the EMBook model for this scenario is as follows:

- $Participant\ 1$ sells $Book\ A$ and buys $Book\ C$ while spending €10 with a final budget of €0,
- $Participant\ 2$ sells $Book\ C, Book\ D$ and buys $Book\ A, Book\ E$ while earning €25 with a final budget of €25,
- $Participant\ 3$ sells $Book\ E$ and buys $Book\ F$ while spending €10 with a final budget of €0,
- $Participant\ 4$ sells $Book\ F$ and buys $Book\ D$ while spending €5 with a final budget of €10

which yields a total transaction volume of €140. As also seen from the example, the model does not allow any participant to have a budget deficit after the market is cleared.

The implementation of the model is also straightforward. Within a predefined time period, sales and purchase requests are collected from the participants. At the end of this period, the market is cleared by solving the winner determination problem which is introduced in the next section. Unsatisfied requests of a participant can be transferred to the next round if the participant wants. The length of the rounds can be

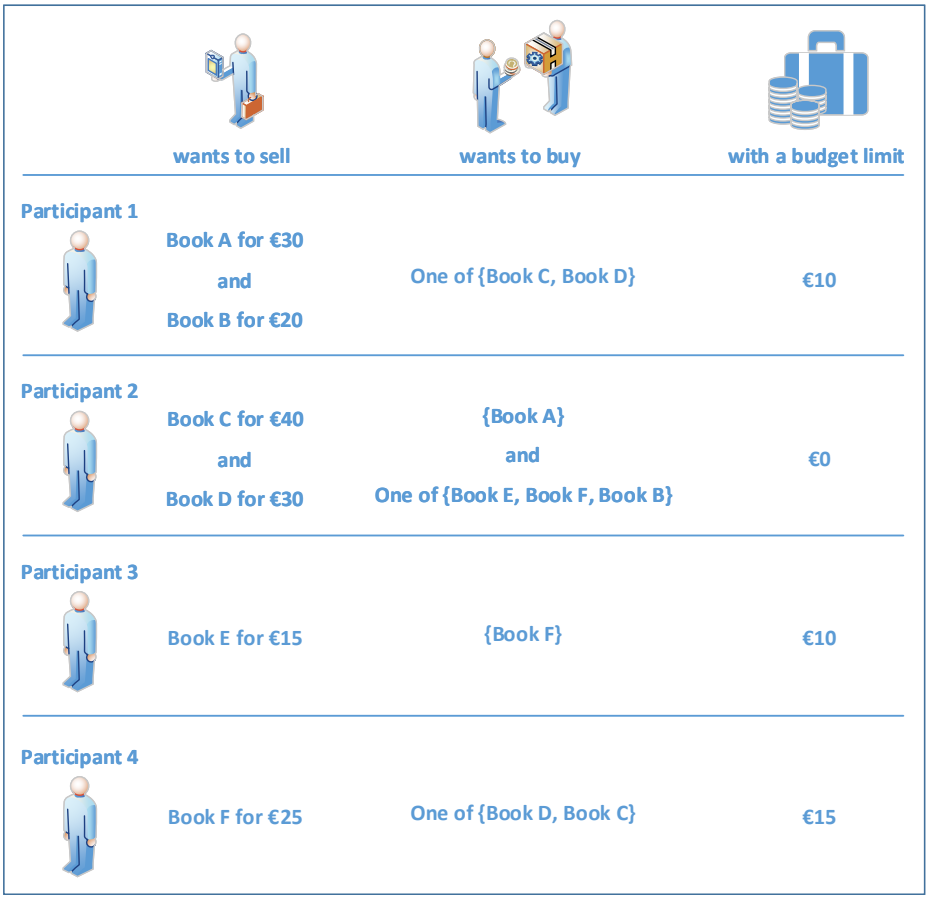

Fig. 1. An example scenario illustrating the EMBook electronic market model for book trading.

determined according to the number of participants and the rate of submission of requests in the market. The longer periods result in better allocative efficiency but they also cause less trading volume to occur per unit time, i.e. reduces the market throughput.

III. Mathematical Definition and Formulation of the EMBook Model

The EMBook model is formally defined as follows: Let $T = \{t_1, t_2, \ldots, t_m\}$ be the set of m participants in the market and B_i be the set of books to be sold by participant t_i ($1 \leq i \leq m$). The set of all books, $B = \{b_1, b_2, \ldots, b_n\}$, is defined as $B = \bigcup_{i=1}^{m} B_i$ $(\forall i, i' \mid B_i \cap B'_i = \emptyset)$. Note that in this model, each book is considered as a unique item. The tuple $P = (p_{b_1}, p_{b_2}, \ldots, p_{b_n})$ denotes the prices of the books where p_{b_j} is the price of the book b_j as declared by its owner ($1 \leq j \leq n$, $p_{b_j} \in \mathbb{R}^+ \cup \{0\}$). The budget limits of the participants are denoted by the tuple $L = (l_1, l_2, \ldots, l_m)$ where l_i is the budget limit of the participant t_i ($l_i \in \mathbb{R}^+ \cup \{0\}$).

In the EMBook model, a purchase request, $r_k = (r_{k1}, r_{k2}, \ldots, r_{kz})$, is an ordered set consisting of z books which are ordered according to the preferences of the request owner ($1 \leq l \leq z, r_{kl} \in B$). That is, $(r_{k1} \succ r_{k2} \succ \ldots \succ r_{kz})$, where $r_{kx} \succ r_{ky}$ means that the request owner prefers book r_{kx} over book r_{ky}. The set of purchase requests submitted by the participant t_i is denoted as R_i, and the set of all purchase requests, $R = \{r_1, r_2, \ldots, r_v\}$, is defined as $R = \bigcup_{i=1}^{m} R_i$.

The meaning of a purchase request can be stated as follows: By submitting a purchase request r_k, the participant t_i declares that she wants to purchase at most one of the books in r_k. The purchase request r_k is called *satisfiable* if there exists at least one book in the purchase request r_k which is available for purchase and the price of the book is within the budget of the participant. The budget of the participant t_i is defined as $proceeds\ of\ the\ sold\ books\ of\ t_i + budget\ limit\ of\ t_i - expenses\ of\ t_i\ for\ purchased\ books$. The *winner determination problem (WDP)* of the EMBook model is defined as finding the maximum cardinality set of mutually satisfiable purchase requests such that the weighted sum of the traded books is maximized.

In order to formulate the problem using linear integer programming, a binary variable x_{kl} is introduced. It denotes whether the book r_{kl} is purchased in the purchase request r_k (1) or not (0). The linear integer programming formulation of the winner determination problem is as follows:

$$\text{First Level: max} \quad \sum_{\substack{r_k \in R \\ r_{kl} \in R_k}} w'_{kl} \cdot x_{kl}, \tag{1}$$

$$\text{Second Level: max} \quad \sum_{\substack{r_k \in R \\ r_{kl} \in R_k}} w''_{kl} \cdot x_{kl} \tag{2}$$

$$\text{s.t.} \quad \sum_{\substack{r_k \in R \\ r_{kl} \in R_k \\ r_{kl} = b_j}} x_{kl} \;\leq\; 1 \quad (b_j \in B) \tag{3}$$

$$\sum_{r_{kl} \in R_k} x_{kl} \leq\; 1 \quad (r_k \in R) \tag{4}$$

$$\sum_{\substack{r_k \in R_i \\ r_{kl} \in R_k}} p_{r_{kl}}\, x_{kl} - \sum_{\substack{r_k \in R \\ r_{kl} \in R_k \\ r_{kl} \in B_i}} p_{r_{kl}}\, x_{kl} \;\leq\; l_i \quad (t_i \in T) \tag{5}$$

$$x_{kl} \in \{0,1\}\ (\forall k, l) \tag{6}$$

where

$$w'_{kl} = \begin{cases} p_{r_{kl}} & \text{if } l = 0 \text{ or } w'_{k(l-1)} > p_{r_{kl}} \\ w'_{k(l-1)} & \text{otherwise} \end{cases}$$

and

$$w''_{kl} = \max_k |r_k| - l$$

In this formulation, Eq.(1) is the first level objective function which maximizes the weighted sum of the traded books according to the weights values w'_{kl}, and Eq.(2) is the second level objective function which again maximizes the weighted sum of the traded books, however, according to the weights values w''_{kl}. The objective functions are hierarchical, that is, the model should be optimized according to the first level objective, and then the second level objective. When optimizing the second level objective, only the solutions that would not degrade the objective value of the first level objective are considered. These two level objective functions cause the total trading volume to be maximized while taking the preferences of the participants in consideration which are declared in their purchase requests. This is achieved by assigning prices of the books as the weight values w'_{kl}, i.e. the weight values for the first level objective function, in order to maximize the total trading volume. However, if a participant prefers a cheap book over an expensive one in her purchase request, assigning the price of the expensive book as the weight value of that book would cause the model to assign the expensive book to the participant even if the cheaper one is also assignable. In order to prevent this kind of situations, the weight values w'_{kl} are determined such as they monotonically decrease for the books requested in the purchase request. Thus, the weight value of the expensive book would be same as the weight value of the cheaper alternative given that the participant prefers the cheaper book over the expensive one. The second level objective function breaks the tie between the books requested in a purchase request in which two or more books exist with the same weight value w'_{kl}.

Regarding the constraints, Eq.(3) ensures that each book can be purchased by at most one participant. Eq.(4) enforces that in each purchase request, at most one book will be purchased by the request owner. Finally, Eq.(5) is the budget constraint, that is for each participant the total cost of the purchased books minus the proceeds of the sold books should not exceed the budget limit of that participant.

The subset sum problem [10, p. 243] can be reduced in polynomial time to the winner determination problem, proving that the winner determination problem is NP-hard. Moreover, when the budget limits of all participants are 0, then the problem also becomes inapproximable. However, it is obvious that if at least one participant has enough budget to purchase at least one of the books in one of her requests, then finding a nonzero feasible solution becomes a polynomial-time problem. Also, at the other end, if budget limits of all participants allow them to purchase every book they want without using the revenue obtained from sold books, then the problem becomes a network problem and thus can be solved in polynomial-time. The proofs for these statements are provided for a similar model in the author's previous work [11].

IV. Solution Methods

Since the winner determination problem of the EMBook model is NP-hard, three heuristic methods were designed. The pseudocode for the first heuristic method, called *Forward-Satisfy (FS)*, can be seen in Alg. 1. In this method, first a list S of subrequests is generated based on the list of all of purchase requests R in the problem instance P. For instance, if a participant's request is $\{Book\ C, Book\ A\}$, two subrequests one for $Book\ C$ and one for $Book\ A$ are included in S. A subrequest is a data structure comprising the owner of the subrequest ($owner$), the requested book ($book$), the index of the subrequest in S ($index$), and the flag indicating whether the subrequest is satisfied or not ($satisfied$). After the list S is generated, all the subrequests in the list is marked as unsatisfied and the list is sorted in descending order according to a given sorting criterion. In this study, four different sorting criteria are tested. These criteria are:

(i) the weights of the subrequests (Weight),
(ii) the prices of the books (Price),
(iii) weight-price ratios (Weight / Price), and
(iv) the weight times price values (Weight * Price).

In these sorting criteria, the value w'_{kl} is used as the weight value for each subrequest. However, if w'_{kl} values are equal for different subrequests, then comparisons are done based on the values w''_{kl} instead.

After the list S is sorted, the first subrequest in the list S (marked as the current subrequest) is checked whether it can be satisfied or not. A subrequest is satisfiable if:

(i) the subrequest is not already satisfied,
(ii) the owner of the subrequest has enough budget to purchase the book requested in the subrequest,
(iii) any other subrequest in the same request is not already satisfied,

Algorithm 1 ForwardSatisfy

Input: An EMBook problem instance P, a $SortingCriteria$ for sorting subrequests
Output: A list S_{sol} of satisfiable subrequests

Generate a list S of subrequests in P.
$S_{sol} \leftarrow \{\}$
Sort S according to $SortingCriteria$
Mark all subrequests in S as $unsatisfied$
$sIndex \leftarrow 0$
while $sIndex < |S|$ **do**
  $retIndex \leftarrow |S|$
  $subReq \leftarrow S[sIndex]$
  if satisfiable($subReq$) **then**
    commit($subReq$)
    $S_{sol}.add(subReq)$
    for all $subReq2$ such that $subReq2.owner = subReq.book.owner$ **do**
      if $(subReq2.index < retIndex)$ **and** $(subReq2.index < sIndex)$ **and** satisfiable($subReq2$) **then**
        $retIndex \leftarrow subReq2.index$
      end if
    end for
  end if
  if $retIndex < |S|$ **then**
    $sIndex \leftarrow retIndex$
  else
    $sIndex \leftarrow sIndex + 1$
  end if
end while
return S_{sol}

(iv) the book requested in the subrequest is not already sold.

If the current subrequest is satisfiable (which is checked using $satisfiable$ method), then it is committed, meaning that the budget of the request owner is decreased and the budget of the book owner is increased by the price of the book. Furthermore, the book requested in the current subrequest is also marked as sold. After that, the minimum index of the satisfiable subrequests of the owner of the book is found and compared to the index of the current subrequest. If the former is smaller, then the algorithm jumps to the former subrequest. If the latter is smaller, or if the current subrequest is not satisfiable at all, then the algorithm moves to the next subrequest in the list S. The algorithm terminates after the list S is traversed to the end.

In the FS method, a subrequest is enabled if the owner of the subrequest has enough budget to purchase the book requested in the subrequest. In the second proposed method, called *ForwardSatisfyWithIncome (FSWI)*, if the subrequest owner has not enough budget to purchase the book, then the method tries to improve the income of the subrequest owner. The pseudocode of the FSWI method can be seen in Alg. 2. Thus, in the FSWI method, $satisfiabilityNBC$ method (NBC stands

Algorithm 2 ForwardSatisfyWithIncome

Input: An EMBook problem instance P, a $SortingCriterion$ for sorting subrequests
Output: A list S_{sol} of satisfiable subrequests

Generate a list S of subrequests in P.
$S_{sol} \leftarrow \{\}$
Sort S according to $SortingCriterion$
Mark all subrequests in S as $unsatisfied$
$sIndex \leftarrow 0$
while $sIndex < |S|$ **do**
  $retIndex \leftarrow |S|$
  $subReq \leftarrow S[sIndex]$
  if satisfiableNBC($subReq$) **then**
    if $(subReq.owner.budget < subReq.price)$ **then**
      $S_{imp} \leftarrow \{\}$
      $budgetFixed \leftarrow false$
      for all $inSubReq$ such that $inSubReq.book.owner = subReq.owner$ **do**
        if satisfiable($inSubReq$) **then**
          commit($inSubReq$)
          $S_{imp}.add(inSubReq)$
          if $subReq.owner.budget \geq subReq.price$ **then**
            $budgetFixed \leftarrow true$
            break {for all loop}
          end if
        end if
      end for
      if not $budgetFixed$ **then**
        rollback(S_{imp})
        $sIndex \leftarrow sIndex + 1$
        continue {while loop}
      else
        $S_{sol}.add(S_{imp})$
      end if
    end if
    commit($subReq$)
    $S_{sol}.add(subReq)$
    for all $subReq2$ such that $(subReq2.owner = subReq.book.owner)$ **or** $(subReq2.owner = subReq.owner)$ **do**
      if $(subReq2.index < retIndex)$ **and** $(subReq2.index < sIndex)$ **and** satisfiable($subReq$) **then**
        $retIndex \leftarrow subReq2.index$
      end if
    end for
  end if
  if $retIndex < |S|$ **then**
    $sIndex \leftarrow retIndex$
  else
    $sIndex \leftarrow sIndex + 1$
  end if
end while
return S_{sol}

for NoBudgetCheck) is used to check the satisfiability of the current subrequest instead of $satisfiability$ method used in the FS method. The $satisfiabilityNBC$ method checks only satisfiability conditions (i), (iii) and (iv) listed above. Then, if the owner of the current subrequest does not have enough budget to purchase the book in the subrequest, the FSWI method tries to improve the budget of the subrequest owner by trying to satisfy incoming subrequests first, that is to commit the subrequests inside which one of the books of the current subrequest owner is requested. If by this process, the budget of the subrequest owner is fixed, then the current subrequest is committed, otherwise all the committed subrequests in this process are rollbacked. The method continues with the next subrequest the index of which is determined in accordance with the smallest index of the satisfiable subrequests of the participants whose budget are increased when the current subrequest is committed as seen in lines 34-43 of Alg. 2.

Both FS and FSWI methods are forward traversing methods which start with an empty solution and construct a feasible solution by trying to satisfy the subrequests in the list S one by one without sacrificing feasibility. In the third proposed method, called *BackwardRollback (BR)*, the reverse approach is taken such that at first all the subrequests are committed producing most likely an infeasible solution. Then, the list of subrequests S is traversed in the reverse direction of the traversal direction of the forward methods. During the traversal, the current subrequest is checked whether it contributes to the infeasibility of the current solution. If so, then it is rollbacked. It may be the case that after the current subrequest is rollbacked, the owner of the book requested in the subrequest may have a budget deficit. If this is the case, then the largest index of the already committed subrequests of the book owner is found. If this index is larger than the index of the current subrequest, this index is used as the index of the next subrequest to be processed. Otherwise, the method moves to the next subrequest in the list S. Note that after S is traversed, it is guaranteed that the BR method produces a feasible solution although in the worst case it may be a zero solution. When a feasible solution is obtained, some participants may have remaining budgets to purchase books in some of their unsatisfied subrequests. In order to satisfy these subrequests, the BR method calls FSWI method as to improve the current feasible solution. The pseudocode of the BR method can be seen in Alg. 3.

The complexity analyses of the proposed heuristic algorithms are quite straightforward. The worst case time complexities of all proposed heuristics are O(n^2) where $n = \max_k |r_k| * |R|$ and space complexities are only O(n).

Algorithm 3 BackwardRollback

Input: An EMBook problem instance P, a $SortingCriterion$ for sorting subrequests
Output: A list S_{sol} of satisfiable subrequests

1: Generate a list S of subrequests in P.
2: $S_{sol} \leftarrow S$
3: Sort S according to $SortingCriterion$
4: **for** $i = 0$ **to** $|S| - 1$ **do**
5: commit($S[i]$)
6: **end for**
7: $sIndex \leftarrow |S| - 1$
8: **while** $sIndex \geq 0$ **do**
9: $retIndex \leftarrow 0$
10: $subReq \leftarrow S[sIndex]$
11: $req \leftarrow$ Index of the Request that $subReq$ belongs
12: **if** ($subReq.satisfied$) **and** (($subReq.owner.budget < 0$) **or** $soldMoreThenOnce(subReq.book)$ **or** $moreThanOneBookPurchasedIn(req)$) **then**
13: rollback($subReq$)
14: $S_{sol}.remove(subReq)$
15: **if** $subReq.book.owner.budget < 0$ **then**
16: **for all** $subReq2$ such that ($subReq2.owner = subReq.book.owner$) **do**
17: **if** ($subReq2.index > sIndex$) **and** ($subReq2.index > retIndex$) **and** ($subReq2.satisfied$) **then**
18: $retIndex \leftarrow subReq2.index$
19: **end if**
20: **end for**
21: **end if**
22: **end if**
23: **if** $retIndex < |S|$ **then**
24: $sIndex \leftarrow retIndex$
25: **else**
26: $sIndex \leftarrow sIndex - 1$
27: **end if**
28: **end while**
29: Call ForwardSatisfyWithIncome with the current solution S_{sol}
30: **return** S_{sol}

V. Experimental Results

In order to estimate the performances of the proposed heuristic methods under real-life market conditions, a test case generator was developed and a test suite was prepared. The test case generator uses GNU Scientific Library [12] for generating pseudo-random numbers which supports all common continuous and discrete random number distributions.

The generated test suite consists of 1600 problem instances in which the number of participants varied between 2,000 and 10,000 for simulating different market sizes. The following parameters of the case generator: the number of books that each participant put up for sale, the number of purchase requests, the number of purchase requests per participant, and the sizes of the purchase requests are configured as to be distributed with Poisson distribution with *mean values* varying between 1 and 7. The requested books in the purchase requests are uniformly selected among all the books. In order to determine the prices of the books, a statistical profile is generated according to the study of Ghose et al. [13] which is based on the sales information in the Amazon.com book marketplace. As discussed in Section II, when all the

participants have zero budget limits, the problem instances are difficult to solve. In fact, these instances would possibly have no nonzero feasible solutions at all. On the other hand, when all the participants have enough budget for all their possible purchases, the problem instances becomes quite easy, requiring polynomial time to be solved. Actually, the market instances in the real life would mostly be in between these two endpoints. Therefore, in order to determine the budget limits of the participants in the generated problem instances, five different budget limit ratios are used varying between 5% to 75%. Using these ratio values, the budget limit l_i for a participant t_i is calculated as:

$$l_i = blr_i \cdot (bl_i^{max} - bl_i^{min}) + bl_i^{min}$$

where blr_i is the budget limit ratio, bl_i^{min} the minimum budget the participant t_i needs in order to be able to purchase the cheapest book listed in her requests if all of her books are sold, and bl_i^{max} is the maximum budget she needs in order to be able to purchase all the books she wants even if none of her books are sold.

The generated test cases were solved using Gurobi mixed-integer programming (MIP) solver version 7 [14] on two 8-cores 3.10 GHz CPUs with 128 GB of memory. The solver was configured to use single thread and a time limit of 60 minutes was defined for each instance. The operating system used was 64 bit Linux. Among the generated 1600 problem instances, the MIP solver found the optimal solutions for 972 instances. For the remaining 628 instances, the solver could not find the optimal solution within the time limit, however, the MIP solver was able to find a nonzero feasible solution for these instances.

Optimally solved instances by the MIP solver were used to measure the quality of the solutions found by the three proposed heuristic methods, FS, FSWI, and BR. For each heuristic method, four different sorting criteria which are explained in Section IV are used. For representing the quality of the solutions, a *goodness* measure is defined such as:

$$\text{Goodness of a Sol.} = \frac{\text{Obj. Val of Heuristic Sol.}}{\text{Optimal Objective Value}} \cdot 100\%$$

The goodness of the solutions found by the proposed heuristic methods and the best solution found by all heuristic methods (Best of All) can be seen in Table I. According to the results, among the four sorting criteria, all three heuristics in which the subrequests are sorted in descending order according to *Weight * Price* values find the best solutions. The results for the sorting criterion *Weight* follows the *Weight * Price* criterion by a close margin. As seen from the results, the sorting criterion to be used is quite important causing up to 5% difference in mean goodness values.

Considering the best performing sorting criterion, that is *Weight * Price*, FSWI method performs better compared with the FS and BR methods. Mean goodness values of the solutions found by the FSWI method is approximately 92.4%, that is within less than 8% of the optimal solutions. The corresponding standard deviation is also small, less than 9%. The lowest goodness value obtained in the FSWI method is approximately 40%. Best solutions found by all three heuristics are also very close to the solutions found by the FSWI method indicating that the FSWI method is almost dominant to other two heuristic methods for the generated test instances. Note that the maximum goodness values for all heuristics are 100%, and therefore these value are not included in Table I for the sake of clarity.

For 628 problem instances among the generated 1600 instances, the MIP solver could only find suboptimal solutions (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 given time limit). For these instances, the proposed heuristics found better solutions on average compared to the solutions found by the MIP solver. The results can be seen in Table II. However, in this case, the goodness values were calculated as the ratio of the objective value of the heuristic solution to the suboptimal solution found by the MIP solver. Thus, goodness values may be higher than 100%. For these instances, the FSWI and the BR methods perform almost equal producing approximately 40% better solutions than the solutions found by the MIP solver on average, and more than 400% better solutions for some specific instances.

The running times of the heuristic methods and the MIP solver for all problem instances can be seen in Table III. All three heuristics are very fast, finding solutions less than 1 second on average whereas the MIP solver requires approximately 1500 seconds for an instance on average. The FSWI method again can be considered the best method in terms of running time compared to the other two methods. The maximum running time of the FSWI method is also very low, which is less than 10 seconds for all sorting criteria.

VI. Discussion and Conclusion

In his open letter on used book sales dated April 14, 2002, Jeff Bezos, CEO of Amazon.com, wrote *"... when a customer sells used books, it gives them a budget to buy more new books."* [15]. However, in current book markets, a participant without a budget for purchasing new books must sell her books first so as to get a revenue, after then she may be able to purchase new books. In this study, an electronic market model, the EMBook model was proposed for trading of used books as well as new ones in order to overcome this issue. In this market model, participants may simultaneously place sale and purchase requests for books allowing participants to spend the revenue to be obtained from the books they want to sell for the books they want to buy. Furthermore, a budget limiting mechanism is also provided such that for each participant, the difference between the cost of purchased books and the revenue of sold books does not exceed the declared budget limit of the participant. This mechanism provides the participants to place purchase requests without being afraid of having a budget deficit in case their books are not sold. Additionally, a participant may also be indifferent to multiple books, for instance, there may be multiple sellers of the same

TABLE I
GOODNESS OF SOLUTIONS FOUND BY THE HEURISTIC METHODS FOR THE OPTIMALLY SOLVED INSTANCES

Sorting Criterion	FS			FSWI			BR			Best of All		
	mean	std	min	mean	std	min	mean	std	min	mean	std	min
Weight	**88.9%**	10.9%	31.3%	**92.2%**	8.5%	36.3%	**91.3%**	8.9%	34.1%	**92.2%**	8.5%	36.3%
Price	**85.8%**	13.3%	27.8%	**89.1%**	11.7%	32.5%	**89.1%**	11.7%	32.5%	**89.2%**	11.6%	32.5%
Weight / Price	**88.4%**	11.1%	31.0%	**91.6%**	8.6%	36.1%	**90.7%**	9.1%	33.9%	**91.6%**	8.6%	36.1%
Weight * Price	**90.1%**	10.4%	32.5%	**92.4%**	8.6%	39.7%	**92.3%**	8.8%	38.8%	**92.5%**	8.6%	39.7%

TABLE II
GOODNESS OF SOLUTIONS FOUND BY THE HEURISTIC METHODS FOR THE SUBOPTIMALLY SOLVED INSTANCES

Sorting Criterion	FS				FSWI				BR				Best of All			
	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
Weight	**119%**	86%	26%	446%	**141%**	108%	**34%**	514%	**140%**	107%	32%	518%	**141%**	108%	34%	518%
Price	**108%**	79%	25%	442%	**120%**	90%	**27%**	494%	**120%**	91%	27%	495%	**121%**	91%	27%	495%
Weight / Price	**116%**	83%	26%	439%	**138%**	105%	**33%**	518%	**137%**	105%	32%	518%	**138%**	105%	34%	518%
Weight * Price	**128%**	96%	27%	463%	**140%**	107%	**34%**	518%	**141%**	107%	34%	521%	**141%**	107%	34%	521%

TABLE III
RUNNING TIMES OF THE HEURISTIC METHODS AND THE MIP SOLVER (IN SECONDS) FOR ALL INSTANCES

Sorting Criteria	FS			FSWI			BR			MIP Solver		
	mean	std	max	mean	std	max	mean	stdev	max	mean	std	max
Weight	**0.3**	0.7	9.4	**0.1**	0.2	2.1	**0.3**	0.6	6.1			
Price	**0.2**	0.5	6.9	**0.2**	0.6	8.2	**0.3**	0.7	10.4			
Weight / Price	**0.2**	0.5	6.1	**0.1**	0.1	1.4	**0.2**	0.4	3.9	**1490**	1733	3600
Weight * Price	**0.4**	0.9	12.8	**0.1**	0.5	6.6	**0.3**	0.6	7.2			

book or the participant may be indifferent to the different editions of a book. For such situations, the participant can declare a set of substitutable books which is ordered according to the participant's preferences. Then, the model ensures that the participant buys at most one of the books from this set. By means of these features, the EMBook model aims to attract more participants to the book markets and to increase the markets' allocative efficiencies.

In this study, the EMBook model was defined mathematically and the winner determination problem of the EMBook model was formulated as a multi-objective linear integer program. Since this problem is NP-Hard, three polynomial-time heuristic methods were also proposed. In order to understand whether the model can be used in large-scale online electronic markets efficiently, a test suite consisting of 1600 test instances with up to 10,000 participants were prepared. These instances were solved using the state-of-the-art MIP Solver and also using the proposed heuristic methods. The MIP solver failed to solve approximately 40% of the instances optimally within one hour of execution time. For the optimally solved instances, the best heuristic method, ForwardSatisfyWithIncome, provided results as good as 92.4% on average with respect to the optimal solutions with a standard deviation of less than 9%. For the remaining instances, this heuristic method provided solutions with 40% better objective values on average compared with the solutions found by the MIP solver in one hour. The proposed heuristics, however, are quite fast requiring less than 1 second on average and less than 10 seconds maximum. The high quality of the solutions found by the proposed heuristic methods and methods' low polynomial complexities enable them to be used efficiently in very large-scale electronic markets with tens of thousands of participants. Note that although this study focuses on secondary book markets, the model is surely applicable to the secondary markets in which other types of durable goods are exchanged.

REFERENCES

[1] M. Grieger, "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. doi: http://dx.doi.org/10.1016/S0377-2217(02)00394-6

[2] "Ebay inc q2 2016 company fast facts," 2016, https://static.ebayinc.com/static/assets/Uploads/PressRoom/eBay-Q22016FactSheet-Investor-Site.pdf, accessed on May 2017.

[3] "Alibaba group, financial and metrics," 2016, http://alibaba.newshq.businesswire.com/press-release/alibaba-group-announces-december-quarter-2016-results, accessed on May 2017.

[4] A. Kambil, P. F. Nunes, and D. Wilson, "Transforming the marketspace with all-in-one markets," *Int. J. Electron. Commerce*, vol. 3, pp. 11–28, 1999. doi: 10.1080/10864415.1999.11518346

[5] J. Y. Bakos, "A strategic analysis of electronic marketplaces," *MIS Q.*, vol. 15, pp. 295–310, 1991. doi: 10.2307/249641

[6] ——, "Reducing buyer search costs: Implications for electronic marketplaces," *Management Science*, vol. 43, no. 12, pp. 1676–1692, 1997. doi: 10.1287/mnsc.43.12.1676

[7] H.-G. Lee, "Do electronic marketplaces lower the price of goods?" *Commun. ACM*, vol. 41, pp. 73–80, 1998. doi: 10.1145/268092.268122

[8] E. Wyatt, "Internet grows as factor in used-book business," 2005, http://www.nytimes.com/2005/09/29/books/29book.html, accessed on May 2017.

[9] Y. J. H. Erik Brynjolfsson 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. doi: 10.1287/mnsc.49.11.1580.20580

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

[11] A. H. Özer, "Auction and barter models for electronic markets," Ph.D. dissertation, Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey, 2011.

[12] "Gnu scientific library," http://www.gnu.org/software/gsl, accessed on May 2017.

[13] A. Ghose, 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. doi: 10.1287/isre.1050.0072

[14] "Gurobi Optimization," http://www.gurobi.com/, accessed on May 2017.

[15] "Jeff Bezos' open letter on used book sales," 2002, http://archive.oreilly.com/pub/wlg/1291, accessed on May 2017.