A Fair, Preference-Based Posted Price Resale E-Market Model and Clearing Heuristics for Circular Economy

Ali Haydar Özer*

Department of Computer Engineering,

Faculty of Engineering,

Marmara University, Istanbul, Turkey.

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

This is the accepted version of the article submitted to Applied Soft Computing journal. This version contains all the materials without professional editing. The published journal article is available at `https://doi.org/10.1016/j.asoc.2021.107308`

Full Citation:

Ali Haydar Özer,

A fair, preference-based posted price resale e-market model and clearing heuristics for circular economy,

Applied Soft Computing,

Volume 106,

2021,

107308,

ISSN 1568-4946,

https://doi.org/10.1016/j.asoc.2021.107308.

*Address: Department of Computer Engineering, Faculty of Engineering, Marmara University, 34722, Istanbul, Turkey. Tel: +90-216-777-3533 E-mail address: haydar.ozer@marmara.edu.tr

A Fair, Preference-Based Posted Price Resale E-Market Model and Clearing Heuristics for Circular Economy

Ali Haydar Özer*

Department of Computer Engineering,

Faculty of 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: +90-216-777-3533 E-mail address: haydar.ozer@marmara.edu.tr

Abstract

Resale markets in which secondhand new and used goods are traded play an important role in circular economy with significant economic and environmental benefits. This study proposes a preference-based posted price electronic market model for resale markets which features several mechanisms to improve the market outcome. The proposed model allows market participants to post sales and purchase orders simultaneously inside a trading round which also enables participants to use the revenue obtained from the items to be sold to purchase other items in the market. Besides, each participant is allowed to declare a budget constraint which restricts the amount that the participant will spend in the market to prevent a possible budget deficiency. Furthermore, the model also allows participants to declare their preferences of substitutabilities in their orders. In this study, the proposed model is formally defined, the corresponding market clearing problem is formulated as a hierarchical multi-objective linear integer program to provide fair allocation between the participants. Four different objective functions are proposed, and their outcomes are compared to the current market system. Since the clearing problem is NP-Hard, several heuristic methods including ant colony optimization, artificial bee colony and genetic algorithms along with problem-specific operators are proposed. The performance of the model is statistically analyzed based on several experiments. The genetic algorithm using the proposed problem-specific operators provides solutions within 3% of the optimal objective values and within 1% of the optimal fairness on average. The results also indicate that the model provides improved market outcomes and fair allocation of items among the participants, and thus it has a potential to contribute to the growth of circular economy.

Keywords: Resale Market; Circular Economy; Ant Colony Optimization; Artificial Bee Colony; Genetic Algorithm.

1 Introduction

Recent advances in information technology have made a major impact on the functions of markets and provided an ability to shift from the traditional physical markets where the traders meet at a certain place and at a certain time for exchanging commodities, to the electronic markets (e-market) [1]. The key feature of an e-market is that it brings multiple buyers and sellers in contact by weakening time and space restrictions [2]. Therefore, an e-market has the potential of attracting more participants than a physical market. For instance, eBay, the world's largest online market, has more than 180 million active users globally [3] and although established in 1995, by 2001 it became the third-ranked website in which the users spend their time online [4]. Another example is Alibaba.com, the world's biggest business-to-business market which has more than 670 million active users [5]. Additionally, e-markets allow increased product variety compared to their physical counterparts. For instance, according to the study of Brynjolfsson et al. [6], Amazon and Barnes and Noble have 2.3 million books listed on their online markets whereas a typical physical 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 items in their online version, walmart.com. Brynjolfsson et al. [6] also show that the gain of the consumers from the increased product variety can be up to 10 times higher than the gain obtained from the price reduction in e-markets. Furthermore, an e-market can also reduce buyers' search costs to obtain information about the product offerings of sellers [7, 8]. This increases the allocative efficiency of the market, i.e. the efficiency with which a market is allocating resources [9].

This study focuses on online resale e-markets for durable goods in which used goods, as well as new ones, are traded. Resale markets for durable goods constitute a significant share in the overall economy. For instance, according to the thredUP 2019 Resale Report, the market size for retail apparel in the US was estimated as \$24 billion in 2018 and expected to grow to \$51 billion in 2023 with an annual growth rate of more than 15% [10]. The increasing popularity of online apparel resale markets such as Depop, Poshmark, and threadUP contribute to this growth. Again, the thredUP 2019 Resale Report states that 56 million women in the US purchased used apparel in 2018 whereas this number was only

4 million in 2017. Note that during the five-year period from 2017 to 2022, retail apparel is expected to grow only 2% annually (from $360 billion to $400 billion) [11]. Another resale market example could be the secondhand luxury goods market with an estimated volume of more than $27 billion globally in 2018 and again the market volume is estimated to increase approximately 12% annually during the 2019 - 2024 period [12]. Furniture is also a popular type of durable goods which can have high resell value. In 2017, the market size for secondhand furniture was estimated at $25 billion globally.

Aside from the contributions to the global economy, resale markets also have significant environmental benefits compared to retail markets through the trading of used goods. Trading of used goods is an integral part of the Circular Economy which is aimed at the reusing of resources to reduce waste [13, 14]. It saves on the number of resources needed to produce the goods that would have otherwise been purchased in the retail market. For instance, growing cotton for producing a pair of jeans requires approximately 7 tons of water which can be saved when the product is reused by another person [15]. Reducing the resources used directly results in reduced carbon emissions. According to the research conducted by Oxfam, in the UK more than two tonnes of apparel are being purchased in the retail market each minute which corresponds to approximately 50 tonnes of carbon being released to the atmosphere at every minute [16]. Another example is Schibsted's Annual Report [17] which estimates that participants of Schibsted's secondhand marketplaces have saved more than 20 million tons of greenhouse gas through used good trading.

The economic and environmental importance of resale markets constitutes the motivation of this study which is to provide a market model for making the resale markets more pervasive to contibute the growth of circular economy. Yet this paper introduces a fair, preference-based posted price electronic market model, called *the PPBM model*, which is designed especially for resale markets. In this model, each market participant can have both seller and buyer roles, that is she can post any number of sales and purchase orders in the market simultaneously inside a single trading round. A sales order includes the description of the item to be sold along with the price that the seller requests. The model considers each item to be sold by sellers as a unique item to allow buyers to distinguish the items to be sold by different sellers. However, if a buyer does not distinguish some items in the market, then the model also allows the buyer to declare a list of alternative items in her purchase order which indicates that the buyer is interested in having only one of these items. For instance, a buyer may not distinguish the edition, condition (e.g. like-new or good) or cover type (hard-cover or soft-cover) of a book, but another buyer may. Furthermore, if a buyer considers several items as alternatives but if she is not indifferent to these items, the model also allows her to declare an order of preference for these items so that if possible, the most preferred item is allocated to her. This mechanism can also be used for sales orders so that the items of a seller are sold in the order of preference declared by the seller.

As mentioned above, the PPBM model is based on the posted-price mechanism so that sellers define the prices of items they want to sell, and the transactions occur at these prices. Since the prices of the items are fixed, there may be multiple buyers who are willing to buy the same items or there may be multiple sellers of items that buyers consider as alternatives. To provide fairness between the participants of the market, the model comprises a mechanism that distributes the total trading volume as evenly as possible between the participants considering their sales and purchase orders.

Finally, since a participant can sell and purchase items simultaneously inside single trading round, the model also enables participants to spend the revenue to be obtained from the items that will be sold for purchasing other items during the market clearing process without requiring any participant intervention. As a part of this mechanism, the participants can also define budget constraints so that the maximum amount of money that each participant may spend in the market is guaranteed to remain within her budget. This mechanism helps to increase the trading volume without requiring a large amount of cash flow in the market and at the same time, it encourages the participants to submit purchase orders freely without a budget deficit risk.

Further details and the benefits of the proposed model are explained in the next section. Section 3 introduces the motivation for implementing the posted price mechanism in the model and provides a comparison to the previous work of the author. In Section 4, the model is formally defined, and the corresponding market clearing problem is defined. This problem is then modeled as a multi-objective linear integer program. Section 5 introduces four different objective functions that can be used directly or extended for different market types. It is shown that the market clearing problem is intractable, and consequently, several heuristic solution approaches are introduced in Section 6. The performances of the proposed solutions are measured on a large test suite, and the statistically analyzed results of these experiments are presented in Section 7. This section also includes the empirical assessment of the outcomes of the model using the proposed objective functions compared to the current market system. Finally, the paper concludes in Section 8.

2 The PPBM Model

In this section, an example market scenario for the PPBM model is going to be introduced, and the market rules of the model are going to be explained in detail. In the example scenario, which is illustrated in Figure 1, there are four participants who post sales and purchase orders for seven items in total. Each sales order consists of the proper description of an item and the price that the seller of the item requests for. For instance, Participant 1 declares that she wants to sell Item A and Item B for €18 and €36, respectively. Participant 3, on the other hand, does not post any sales orders, meaning that she has only a buyer role in the market. In addition to the sales orders, each seller may also declare an order of preference for the items she wants to sell in the market. For instance, In the examples scenario Participant 1 declares the following order of preference for her items:

$$\text{Item } B \succ \text{ Item } A$$

which means that she prefers, of course, all two items to be sold in the market, however, if only one could be sold, she prefers that Item B to be sold instead of Item A. Similarly, Participant 2 states that if only one of her items could be sold, then she prefers Item C to be sold. If two of her items could be sold, then she prefers Item C and Item E over Item D.

A purchase order in the model consists of a list of a non-empty subset of items available in the market which indicates that the participant who posts the purchase order wants to buy one of the items in that list. For instance, Participant 1 declares that she wants to buy one of the items D, E, and F. However, she also submits a second purchase order for Item G, indicating that she also wants to purchase Item G in addition to one of the

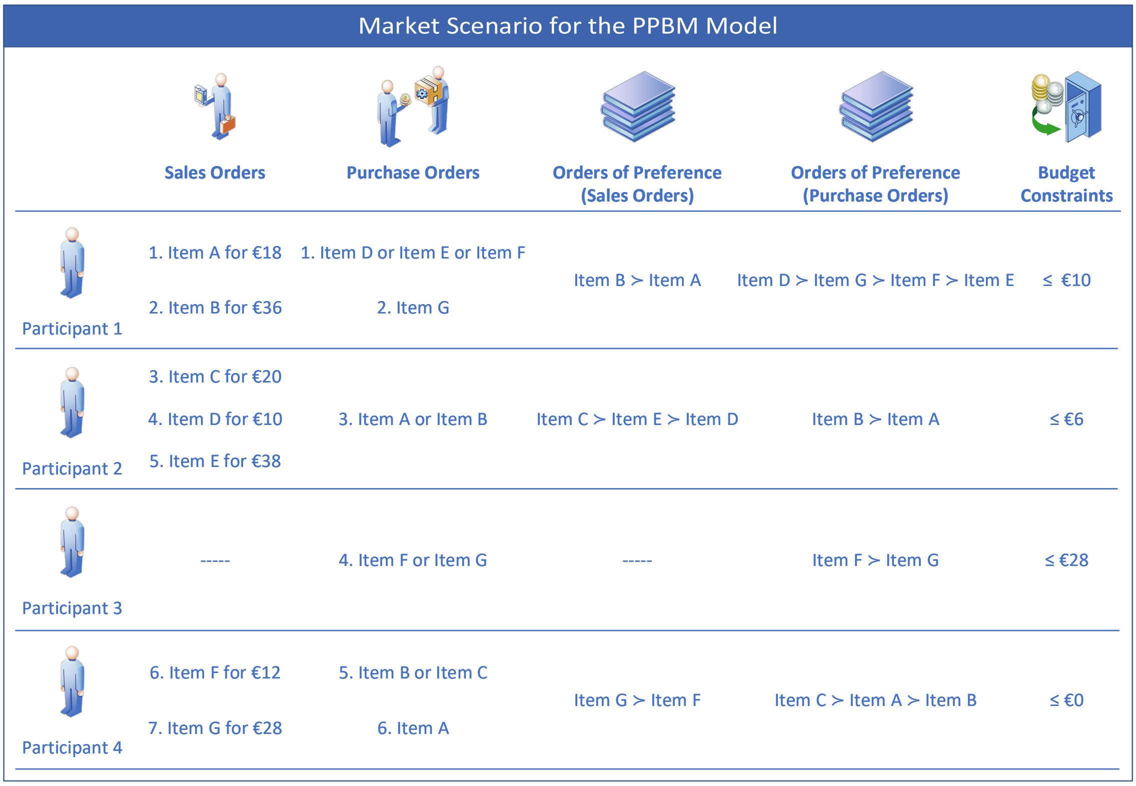

Figure 1: An example scenario illustrating the PPBM model.

items D, E, and F. Thus, when the market is cleared, she can purchase at most two items. A participant can post any number of purchase orders. Similar to the sales orders, each participant can declare an order of preference for the items she wants to purchase. For instance, as explained above Participant 1 is interested in four items, D, E, F, and G, among which she wants to purchase at most two items. She declares the following order of preference:

$$\text{Item } D \succ \text{Item } G \succ \text{Item } F \succ \text{Item } E$$

indicating that she prefers to purchase Item D over Item G, Item G over Item F, and so on. Thus, she would be highly satisfied if she can purchase Item D and Item G in the market.

If a participant is indifferent to two or more items in her purchase orders, any default order of these items (e.g. based on the system assigned id of items) can be used. Note that although a purchase order limits the number of items to be purchased to one, this does not violate the generality of the model. If a participant wants to purchase more than one item in her purchase order, the corresponding purchase order can be duplicated as many times as necessary. Purchase orders of a participant are not necessarily unique in this model and a participant can post any number of purchase orders.

Unless identical instances of a product are sold by a single seller, buyers in the market may differentiate the instances posted on the market based on several factors such as the location of the seller, the reputation of the seller, shipping fees and delivery times, conditions of used items, editions/versions of items, available payment options, and the return policy. To handle different preferences of the participants, the model lets the participants with buyer roles decide on the substitutability of items. Therefore, each item posted in the market is treated as a unique item even if there are multiple instances of the same product exist in the market. However, as described above, the model also allows a buyer to group a number of items inside a purchase order to indicate that she considers these items substitutable. If the buyer is not indifferent to these items, the model also allows the buyer to prioritize these items by declaring an order of preference.

In addition to sales and purchase orders, each participant can further declare a budget constraint covering all her transactions in the market. For instance, in the example scenario, Participant 1 declares that she wants to spend at most €10 in the market. This limit restricts the trade so that the participant can only buy Item D unless her items are sold first. Participant 4, on the other hand, declares a zero budget limit meaning that she would be able to purchase one or more items in the market only if at least one of her items is sold.

The PPBM model being a two-sided market model in the sense that each participant can post both sales and purchase orders inside a trading round, introduces the possibility of using the revenue of a participant in the market as a budget for purchasing new items. In the current market system, if a participant does not have enough money to purchase the items she wants, she has to first sell one or more of her items to obtain the required budget for purchasing. This requirement would prevent potential trading to occur between the participants with limited budgets in the current market system. However, in the PPBM model, trading patterns including the participants with limited budgets can be found and initiated. The benefit of this feature can be seen in the example scenario. Figure 2 depicts the outcome of the current market system. Based on the declared budget limits of

Market Outcome of the Current Market System			
	Sells	**Buys**	**Balance**
Participant 1	-----	Item D for €10	€0
Participant 2	Item D for €10	-----	€16
Participant 3	-----	Item F for €12	€16
Participant 4	Item F for €12	-----	€12

Figure 2: Market outcome of the current market system for the example scenario illustrated in Figure 1. Note that the balance values include the initial balance values determined by the budget limits of the participants.

the participants, only two participants can purchase items in the current market system. Participant 1 purchases Item D for €10 and since Participant 3 prefers Item F over Item E, she purchases Item F for €12. Thus, two items are exchanged in the market with a total volume of €22.

In the market outcome of the PPBM model which can be seen in Figure 3, however, six out of seven items are traded with a total trading volume of €124 which demonstrates the benefit of the model. The budget constraint mechanism ensures that none of the participants has a budget deficit, i.e. negative balance, at the end of the market clearing process.

The implementation of the market mechanism may vary according to the market type, however, in a typical implementation, there can be three periods inside a trading round. In the first period, the participants post their sales orders and their orders of preference to the system. Thus, at the end of the first period, the information for the items to be sold in the market becomes available. In the second period, the purchase orders and orders of preference are collected from the participants, the participants may also modify or retract their purchase orders during this period. In the third period, the market is cleared using the market clearing process which is described in Section 4. The orders which are failed to be satisfied can be transferred to the next round of the market depending on the participants'

Market Outcome of the PPBM Model			
	Sells	**Buys**	**Balance**
Participant 1	Item A for €18 Item B for €36	Item D for €10 Item G for €28	€26
Participant 2	Item C for €20 Item D for €10	Item B for €36	€0
Participant 3	-----	Item F for €12	€16
Participant 4	Item F for €12 Item G for €28	Item C for €20 Item A for €18	€2

Figure 3: Market outcome of the PPBM model for the example scenario illustrated in Figure 1. Note that the balance values include the initial balance values determined by the budget limits of the participants.

requests. The length of the periods should be determined based on the number of sales and purchase orders collected. The longer periods provide increased allocative efficiency while reducing market throughput.

3 Auctions vs. Posted Prices

Typically, e-markets provide one or both of the following mechanisms for the trading of items: auction mechanism and posted-price (fixed-price) mechanism. For instance, eBay offers both of these mechanisms, therefore it becomes an issue for the sellers to decide which mechanism to be used for selling their items. There are many studies even before the Internet era indicating that auctions provide higher expected revenue to the sellers (see, for example, the work of Bajari and Hortaçsu [18] for a review). Economist [19] predicted that the Internet would be "creating the possibility of a permanent worldwide bazaar in which no prices are ever fixed for long, all information is instantly available, and buyers and sellers spend their lives haggling to try to get the best deals". Therefore, auctions were the dominant preferred mechanism in the early 2000's among the sellers on eBay. The proxy bidding mechanism offered by eBay also contributed to this high usage of the auction mechanism which allows buyers to set a maximum price of their bids, and let the system adjust the price of a bid automatically in case another buyer offers a higher price. This mechanism reduces the necessity for the bidders to pay continuous attention to the auction [4].

However, the mechanism choices of sellers for their listings have changed dramatically since then. Einav et al. [4] has analyzed the propriety data obtained from eBay starting from January 2003 till January 2016 to present the average daily share of auction-based listings to all listings, i.e. the number of both auction-based and posted price listings. At the beginning of 2003, the share of auction-based listings was higher than 95%. This ratio gradually decreases to less than 10% over the years. At the beginning of 2016, more than 90% of all listings posted on eBay were posted price listings. This observation motivated some researchers to further analyze sellers' behaviors. Zeithammer and Liu [20] analyzed Canon digital camera listings on eBay in September and October 2005 and found out that sellers with large stocks mostly prefer posted price listings. Hammond conducted a similar study for compact disc sales on eBay [21, 22]. He concluded that the decision of mechanism selection of a seller depends on her outside options. Sellers with higher selling opportunity cost favor posted price listings, and sellers with less opportunity cost favor auctions. Bauner [23] analyzed eBay listings for Major League Baseball Tickets. His findings indicate that sellers would prefer a market with only posted price listings allowed over the markets supporting both posted price and auction listings. The reason is that sellers' surplus in posted price only market would be 6.4% higher than in markets supporting both mechanisms. Further discussion on this subject can be found in [4].

The recent trend that the sellers prefer the posted price mechanism over the auction mechanism is the main reason for proposing a posted price-based model in this study. In the previous work [24], the author of this paper proposed a double auction-based market model, called the DABC model, considering the theoretical advantages of the auction institution. Similar to the PPBM model, the DABC model includes a budget constraint mechanism so that the bidders in the DABC market cannot have a budget deficit. However, the pricing mechanisms used in these models differ significantly. In the DABC model, each seller submits one or more asks which include the item to be sold and the associated *ask price* which indicates *the minimum price* she is willing to sell the item. Every buyer, on the other hand, submits one or more bids including an item to be purchased and the associated *bid price* which indicates *the maximum price* the buyer is willing to buy the item for. The utility of a transaction is defined as the difference between the bid and ask prices for an item. The transaction occurs at a price that is defined by the k-DA policy [25]. The DABC model has a single objective function which is to maximize the total utility of the bidders. The PPBM model which is proposed in this study is a multi-objective model that includes a mechanism for participants to declare their preferences for the items they want to sell and purchase and a further mechanism to provide fairness between the market participants. Since the market clearing problems are also different for both models, the experiments and the corresponding market clearing algorithms are also substantially different.

However, both models complement each other to attract more participants to a resale market. As eBay offers both auction and posted price mechanisms to its users, a resale market maker can implement both the DABC and the PPBM models and offer both mechanisms to the market participants.

Finally, although the proposed PPBM model can be said to utilize a hidden bartering mechanism through enabling the income of the items sold in the market to be used for purchasing another items through the budget constrain mechanism, the mechanism of the PPBM model is different from that of the pure bartering based models introduced in [26–28] which aim to extract barter cycles in the market. In the PPBM model, since an item is not exchanged (bartered) for another item, only budget constraints are applied, and pure barter cycles are not found. This also reduces the difficulty of finding "double coincidence of wants" which is observed in direct bartering mechanisms [29].

4 Mathematical Definition and Formulation of the PPBM Model

The PPBM model can be defined formally as follows: let

- $H = \{h_1, h_2, \ldots, h_m\}$ be the set of m participants in the market;
- S_i be the set of sales orders that a participant h_i posts in the market, where a sales order $s_{ij} \in S_i$ is a two tuple, $s_{ij} = (t_{ij}, p(t_{ij}))$, in which t_{ij} is the item put for sale and $p(t_{ij})$ is the price of the item t_{ij} ($1 \leq i \leq m$, $1 \leq j \leq |S_i|$);

- $TS_i = \{t_{i1}, t_{i2}, \ldots, t_{iu}\}$ be the preference-ordered set of all items to be sold by the participant h_i which are extracted from S_i. The order of TS_i is defined as ($t_{i1} \succ t_{i2} \succ \ldots \succ t_{iu}$), and $t_{ix} \succ t_{iy}$ means that the participant h_i prefers the item t_{ix} to be sold rather than the item t_{iy} if only one item can be sold ($1 \leq i \leq m$, $u = |S_i|$, $1 \leq (x, y) \leq u$)

- O_i be the set of purchase orders that the participant h_i posts in the market, where a purchase order $o_{ik} \in O_i$ is defined as a set of z items, $o_{ik} = \{t_{ik1}, t_{ik2}, \ldots, t_{ikz}\}$, such that the participant h_i is willing to buy at most one of the items in the set o_{ik}.

- $TP_i = \{t_{i1}, t_{i2}, \ldots, t_{iv}\}$ be the preference-ordered set of all items to be purchased by the participant h_i which are extracted from O_i. The order of TP_i is defined as ($t_{i1} \succ t_{i2} \succ \ldots \succ t_{iv}$), and $t_{ix} \succ t_{iy}$ means that the participant h_i prefers to purchase the item t_{ix} rather than the item t_{iy} ($1 \leq i \leq m$, $v = \sum_{i,k} |o_{ik}|$, $1 \leq (x, y) \leq v$)

- $O = \{o_1, o_2, \ldots, o_y\}$ be the set of all purchase orders in the market which is defined as $O = \bigcup_{i=1}^{m} O_i$ ($y = |O|$);

- finally, m_i be the maximum amount of money that the participant h_i wants to spend in the market ($m_i \in \mathbb{R}^+ \cup \{0\}$, $1 \leq i \leq m$).

The budget B_i of a participant h_i is then defined as:

$$B_i = r_i - e_i + m_i \tag{1}$$

where r_i is the total revenue of the participant obtained from the items sold, and e_i is the total expenses of the participant for the items purchased in the market.

A purchase order o_{ik} of a participant h_i is called *executable* if

- there exits at least one item, $t_{ikx} \in o_{ik}$, which is currently not sold;

- the cost of the item, t_{ikx}, is within the budget of participant h_i ($p(t_{ikx}) \leq B_i$);

- the participant h_i has not yet purchased any other item $t_{iky} \in o_{ik}$ for this purchase order (note that an item can be requested in more than one purchase order of a participant).

Given these definitions, the *market clearing problem (MCP)* of the PPBM model is defined as finding the maximum cardinality set of mutually executable purchase orders such that the weighted sum of the executed sales/purchase orders are maximized.

The MCP is formulated as a linear integer program by introducing the following binary variable:

$$x_{ikx} = \begin{cases} 1 & \text{the item } t_{ikx} \text{ is purchased in the purchase order } o_{ik} \text{ of} \\ & \text{the participant } h_i \\ 0 & \text{otherwise} \end{cases}$$

The linear integer program of the MCP is as follows:

$$\text{Primary Objective: } \max z = \sum_{h_i \in H} \sum_{o_{ik} \in O_i} \sum_{t_{ikx} \in o_{ik}} w_{ikx} \, x_{ikx} \tag{2}$$

$$\text{Secondary Objective: } \min \sum_{h_i \in H} \left| \frac{z^*}{|H|} - \sum_{o_{ik} \in O_i} \sum_{t_{ikx} \in o_{ik}} w_{ikx} \, x_{ikx} \right| \tag{3}$$

$$\text{s.t.} \sum_{t_{ikx} \in o_{ik}} x_{ikx} \leq 1 \quad (h_i \in H, o_{ik} \in O_i) \tag{4}$$

$$\sum_{h_{i'} \in H} \sum_{o_{i'k} \in O_{i'}} \sum_{t_{ij} \in o_{i'k}} x_{ikx} \leq 1 \quad (h_i \in H, (t_{ij}, p(t_{ij})) \in S_i) \tag{5}$$

$$r_i - e_i + m_i \geq 0 \quad (h_i \in H) \tag{6}$$

$$\sum_{h_{i'} \in H} \sum_{o_{i'k} \in O_{i'}} \sum_{\substack{t_{i'kx} \in o_{i'k} \\ \wedge \\ (t_{i'kx}, p(t_{i'kx})) \in S_i}} p(t_{i'kx}) \, x_{i'kx} = r_i \quad (h_i \in H) \tag{7}$$

$$\sum_{o_{ik} \in O_i} \sum_{t_{ikx} \in o_{ik}} p(t_{ikx}) \, x_{ikx} = e_i \quad (h_i \in H) \tag{8}$$

$$x_{ikx} \in \{0, 1\} \tag{9}$$

$$e_i, r_i \geq 0 \tag{10}$$

In this formulation, Eq.(2) is the primary objective function that maximizes the weighted sum of the executed orders based on the weight values w_{ikx}. By using different weight values w_{ikx}, different objective functions can be defined for the PPBM model, four of which are explained in the next section. It is used to maximize the market throughput while preferably considering the preferences of the participants declared in their purchase orders. However, since the prices of the items are fixed and each participant can submit any number of purchase orders in the PPBM model, there may be different market outcomes that give the same optimal primary objective value. Consider the following scenario which illustrates this issue. Suppose that

- Participant 1 sells two items, Item A and Item B;
- Participant 2 wants to purchase both items A and B by placing two purchase orders;
- Participant 3 also wants to purchase both items A and B by placing additional two purchase orders;
- both Participant 2 and Participant 3 have enough budget to purchase both items A and B.

It is obvious that there are four different outcomes of this scenario:

(i) Participant 2 purchases both items A and B;

(ii) Participant 3 purchases both items A and B;

(iii) Participant 2 purchases both Item A and Participant 3 purchases Item B;

(iv) Participant 2 purchases both Item B and Participant 3 purchases Item A;

The outcomes (i) and (ii) are not considered fair since only one participant is satisfied. Therefore, to select the fairest allocation (allocation (iii) or (iv) in this scenario [1]) among the optimal allocations with respect to the primary objective value, a secondary objective is introduced in Eq.(3). This objective is used to minimize the sum of the absolute deviation of each participant's contribution to the primary objective value from the mean contribution of participants. Hence, the secondary objective aims to distribute the optimal primary objective value (z^*) between the participants as evenly as possible to provide fairness between the participants. Note that these two objective functions are hierarchical, that is, the model should be optimized according to the primary objective function first.

[1] Depending on the preferences of participants 2 and 3, allocation (iii) can be preferred over allocation (iv) or vice-versa. If both participants have the same preferences for items A and B, then both allocations would be equally preferrable.

When the optimal primary objective value z^* is found, the model then should be optimized according to the secondary objective function without degrading the optimal primary objective value z^*. In other words, in the first optimization step, a program consisting of the primary objective function Eq.(2) and the constraints Eq.(4-10) is optimized to find z^* . Then, in the second optimization step, another program consisting of the secondary objective function, the constraints Eq.(4-10), and a further constraint

$$z^* \cdot tol \leq \sum_{h_i \in H} \sum_{o_{ik} \in O_i} \sum_{t_{ikx} \in o_{ik}} w_{ikx} \, x_{ikx} \tag{11}$$

is optimized. Note that after the first optimization step, none of the decision variables are fixed. A constant *tol* factor that is close to 1 (e.g. a value of 0.9999) can be used to prevent possible floating-point arithmetic errors which may cause the optimizer to falsely return an infeasible solution error.

The constraints of the model are defined as follows. In the PPBM model, at most one item can be purchased in every purchase order and this constraint is indicated in Eq.(4). Eq.(5) prevents an item to be sold to more than one participant. Eq.(6) is the budget constraint which ensures that no participant has a budget deficit. Eq.(7-8) define the total revenue and the total expenses of each participant, respectively.

The Bank Clearing Problem (BCP) introduced in [30] is a specialization of the MCP of the PPBM model (a BCP instance can be reduced in polynomial time to a limited MCP instance such that only one item is requested in every purchase order, and each item is requested in at most one purchase order). Since the BCP is proven to be NP-hard [30], the MCP of the PPBM model is also NP-hard (in other words, the decision version of the MCP is NP-complete). Even with only two participants, the MCP remains intractable.

5 Determining the Weight Values - Objectives of the PPBM Model

As explained in the previous section, the primary objective of the PPBM model as seen in Eq.(2) introduces the weight values w_{ikx} for each order to maximize the weighted sum of the executed orders. Depending on these weight values, different objective functions can be defined for the PPBM model. In this work, four different objective functions are introduced:

(i) **Maximize the Number of Traded Items (PPBM-Items Traded):** This objective function aims to maximize the number of items traded in the market, therefore every weight value w_{ikx} is set to a constant value, e.g. to 1. So,

$$w_{ikx} = 1 \quad (\forall i, k, x). \tag{12}$$

Since in each purchase order only one item can be traded, each item traded means one purchase order and one sales order are executed. Therefore, this objective function also maximizes the number of orders executed. Note that this function treats all items to be sold and purchased equally.

(ii) **Maximize the Trading Volume (PPBM-Trading Vol.):** This objective function aims to maximize the trading volume of the market which is the sum of the prices (or revenues, since each item in the market is considered as unique) of the exchanged items in the market. Therefore, each weight value w_{ikx} is defined as the corresponding price of the item t_{ikx}. So,

$$w_{ikx} = p(t_{ikx}) \quad (\forall i, k, x) \tag{13}$$

Note that for a participant, this objective function prefers the most expensive item to be sold or purchased by her to the rest of the items listed in her sales or purchase orders given that she has enough budget.

(iii) **Maximize the Preference Values Sum of Items Traded (PPBM-Pref. Sum):** Previous two functions aim to maximize either the total number of items traded or the total trading volume of the market. Thus, they do not consider the order of preferences of the participants for their sales and purchase orders.

This objective function, on the other hand, aims to maximize the total number of items traded based on the preferences of the participants. To achieve this, each weight value w_{ikx} is defined as follows:

$$w_{ikx} = SPV(t_{ikx}) + PPV(t_{ikx}) \tag{14}$$

where $SPV(t_{ikx})$ is the Sales Preference Value of the item t_{ikx} which is defined as

$$SPV(t_{ikx}) = \max_i |TS_i| - r_{TS_i}(t_{ikx}) + \sigma, \tag{15}$$

and $PPV(t_{ikx})$ is the Purchase Preference Value of the item t_{ikx} which is defined as

$$PPV(t_{ikx}) = \max_i |TP_i| - r_{TP_i}(t_{ikx}) + \sigma. \tag{16}$$

Note that $r_{TS_i}(t_{ikx})$ and $r_{TP_i}(t_{ikx})$ are the ranks of the item t_{ikx} in the ordered sets TS_i and TP_i, respectively, and $\sigma > 0$ is the shift parameter. Recall that TS_i and TP_i are the ordered sets of items that a participant h_i wants to sell and purchase respectively ordered according to the preferences of the participant h_i. Thus, the ranks of the first elements in the sets TS_i and TP_i are 1 and the ranks of the last elements are $|TS_i|$ and $|TP_i|$, respectively. The shift parameter σ ensures that the sales and purchase preference values of the least preferred items are higher than 0. A typical value of 1 for σ is used in this study which ensures that the preference value of the least preferred item is 1.

(iv) **Maximize the Blending of Trading Volume and Preferences of the Participants (PPBM-Blended Obj.):** This objective function blends two possibly conflicting objectives in a single objective function which are the total trading volume and the sum of the preference values of the participants. Thus, in this objective function, each weight value w_{ikx} is defined as follows:

$$w_{ikx} = \alpha \, \frac{p(t_{ikx})}{\overline{p}} + (1-\alpha) \, \frac{SPV(t_{ikx})/\overline{SPV} + PPV(t_{ikx})/\overline{PPV}}{2} \tag{17}$$

where α is the blending factor, $\overline{p}$, $\overline{SPV}$, and $\overline{PPV}$ are the mean price, mean sales preference value and mean purchase preference value of the items, respectively. Since the range of the prices of the items and the range of the sales and purchase preference values of the items can be different, normalized price and preference values are used in this objective function. A typical value of 0.5 is used for α in this study which provides equal importance to the total trading volume and the sum of the preference values of the participants.

Note that each proposed primary objective function provides possibly different allocation of items in the market given the same set of posted sales and purchase orders. The allocations provided by these objective functions are compared empirically in Section 7. Depending on the market type, i.e. the type of the items sold in the market, price distribution of the items, preferences of the participants, etc. one of these objective functions or one of their variants can be used.

6 Solution Methods

In Section 4, the market clearing problem (MCP) was defined and formulated using linear integer programming. Thus, a general-purpose mixed-integer programming solver can be used to solve the MCP instances of the PPBM model. However, again as shown in the same section, the MCP is an NP-hard problem, therefore it is not always possible to find the optimum solution or even a feasible solution in a limited time. For this reason, several local search (Hill Climbing and Iterated Local Search) and population-based (Ant Colony Optimization, Artificial Bee Colony and Genetic Algorithms) heuristic methods with varying complexities are proposed in this study for the MCP.

The proposed local search methods require a neighborhood relation to be defined for searching the solution space systematically. Therefore, this study also introduces two problem-specific neighborhood relations designed for the MCP. Besides the local search methods, these relations also used inside the operators of the population based heuristic algorithms. Therefore, before explaining the heuristic methods, first, the neighborhood relations will be explained.

6.1 Neighborhood Relations for the MCP

To understand the proposed neighborhood relations, the definition of suborder should be given first. In the PPBM model, each participant may post any number of purchase orders each of which is a set of one or more items listed in the market. A *suborder* is a partition of a purchase order in which only one item is requested. For instance, in the example scenario in Figure 1, Purchase Order 1 consists of three suborders, one for Item D, Item E, and Item F, and Purchase Order 2 has only one suborder for Item G. A suborder is considered as a data structure which comprises the item requested, the seller of the item requested, the purchase order to which the suborder belongs, the owner of the purchase order, and a flag indicating whether the suborder is executed or not.

Note that since in a suborder only one item is requested, each suborder is directly associated with a single sales order through the item requested in the suborder. Thus, executing a suborder means that the corresponding item is purchased by the participant who posts the suborder, and hence the corresponding sales order of the seller is also executed. Therefore, in the proposed solution methods, only the suborders of the purchase orders are considered. Also note that the PPBM model enforces that at most one of the suborders in a purchase order can be executed, that is executing a purchase order means that executing one of the suborders of the purchase order.

Similar to the notion of executability for purchase orders, a suborder o is called *executable* if

(i) the suborder o is not already executed;

(ii) the item requested in the suborder o is available (i.e. not sold);

(iii) there is no already executed suborder in the corresponding purchase order (i.e. the purchase order to which the suborder o belongs).

(iv) the participant who posted the suborder o has enough budget to purchase the item requested in the suborder o.

When a suborder o is *executed*,

(i) the suborder o is marked as executed;

(ii) the item requested in the suborder o is marked as sold;

(iii) the budget of the buyer is decreased by the price of the item requested in the suborder o, and the budget of the seller is increased by this amount.

Conversely, when a suborder o is *undone*, all the execution steps are undone as follows:

(i) the suborder o is marked as not executed;

(ii) the item requested in the suborder o is marked as available;

(iii) the budget of the seller is decreased by the price of the item requested in the suborder o, and the budget of the buyer is increased by this amount.

Note that the budget of a participant is initialized to the maximum amount of money that the participant wants to spend in the market at the beginning (see Section 4 for the definition of the budget of a participant).

As introduced in Section 4, the integer programming formulation of the MCP requires the decision variable x_{ikx} which denotes whether x^{th} item in the k^{th} purchase order of the participant h_i is purchased by this participant or not. Given the definition of suborder, it is seen that x_{ikx} actually denotes whether the x^{th} suborder of k^{th} purchase order of the participant h_i is executed ot not. Thus, every solution of an MCP instance is a set of executed suborders, and it can be represented as a binary string containing one bit for each suborder which indicates whether the corresponding suborder is executed or not.

Based on this representation of a solution, a neighborhood relation can be defined quite easily: a solution Sol' is a *neighbor* of another solution Sol if only one bit is flipped in the representation of the solution Sol, that is one of the suborders of Sol' is undone if it was already executed in Sol, or executed if it was not executed in Sol. Although the definition seems straightforward to implement, there is an important issue to be handled such that executing or undoing a suborder may result in an infeasible solution. Therefore, two neighborhood methods are proposed which find only feasible neighbors of given solutions.

The pseudocode of the neighborhood method N_1 is presented in Algorithm 1. Given a solution Sol and a suborder o_{chosen} which is to be flipped (i.e. executed or undone), the method first checks whether the suborder o_{chosen} is already executed or not. If it is executed, the method *undoes* the suborder o_{chosen} the steps of which are explained above. However, this action may cause a budget deficiency for the owner of the item since her item would be unsold. If this is the case, then, the method tries to close the deficit of this participant (the owner of the unsold item) by executing the suborders in which an item of this participant is requested. These steps are summarized as *executeOrders* method in Algorithm 2. Note that *executeOrders* method executes the suborders of the owner of the unsold item in the descending order of weight values, that is the method first tries to execute the order with the highest weight value. Also, the method stops executing the suborders when the deficit is closed. If the deficit is closed, the neighborhood method N_1 returns the new solution Sol' as the neighbor solution of given solution Sol. However, if the deficit cannot be closed, the neighborhood method returns $neighborNotFound$ meaning that it is not possible to obtain a neighbor of the given solution Sol by undoing the suborder o_{chosen}.

Input: The initial solution Sol, the suborder o_{chosen} to be executed or undone.
Output: The neighbor solution Sol' or $neighborNotFound$.

Set $Sol' \leftarrow Sol$
if the suborder o_{chosen} is already executed **then**
    Undo the suborder o_{chosen}
    if the budget of the participant h_x who owns the item requested in the suborder $o_{chosen} < 0$ **then**
        {Improve the budget of the participant h_x if possible}
        Set $budgetInDeficit$ ←executeOrders($Sol', h_x, o_{excluded} = o_{chosen}$)
        if $budgetInDeficit$ **then**
            return $neighborNotFound$ {Unable to close the budget deficit}
        end if
    end if
    return neighbor solution Sol'
else if the suborder o_{chosen} is *executable without budget control* **then**
    if the budget of the participant h_y who posted the suborder $o_{chosen} < 0$ **then**
        {Improve the budget of the participant h_y if possible}
        Set $budgetInDeficit$ ←executeOrders($Sol', h_y, o_{excluded} = NULL$)
        if $budgetInDeficit$ **then**
            return $neighborNotFound$ {Unable to close the budget deficit}
        end if
    end if
    Execute the suborder o_{chosen}
    return the neighbor solution Sol'
end if
return $neighborNotFound$ {o_{chosen} is not executable}

Algorithm 1: Pseudocode for Neighborhood Method N_1

However, if the suborder o_{chosen} is not already executed, then the neighborhood method N_1 *tries to* execute the suborder o_{chosen}. As noted above, to execute a suborder, conditions (i)-(iv) should be satisfied. The method first checks whether the conditions (i)-(iii) hold and if at least one of them is not satisfied, then the method fails to return a neighbor solution. However, if the budget constraint, i.e. the condition (iv) does not hold, the method first executes the suborder o_{chosen} which causes a deficit in the budget of the participant who posted the suborder o_{chosen}. Then, the method *tries to* close the budget deficit of this participant through the method *executeOrders* which is described above. If it succeeds, it returns the neighbor solution Sol'.

Input: The solution Sol', the participant h_x whose budget is in deficit, the suborder $o_{excluded}$ to be excluded

Output: The boolean flag *budgetInDeficit* indicating whether the budget of the participant h_x remains in deficit or not

```
Set budgetInDeficit ←false
if the budget of the participant h_x < 0 then
   {Improve the budget of the participant h_x by selling her items}
   Set budgetInDeficit ←true
   Build the list L_x of suborders that request an item sold by the participant h_x
   Sort L_x according to weight values w of suborders in descending order
   for all suborder o_x in L_x do
      if o_x = o_excluded then
         continue
      end if
      if the suborder o_x is executable and budgetInDeficit then
         Execute the suborder o_x
         if the budget of the participant h_x ≥ 0 then
            Set budgetInDeficit ←FALSE
         end if
      end if
   end for
end if
return budgetInDeficit
```

Algorithm 2: Pseudocode for executeOrders Method

The pseudocode for the neighborhood method N_2 is presented in Algorithm 3. The first part of the second neighborhood method N_2 is same as N_1 so that the method first checks whether the suborder o_{chosen} is already executed or not. If it is executed, the method undoes the suborder o_{chosen} and tries to close the possible budget deficit of the owner of the item requested in the suborder o_{chosen}. However, if the suborder o_{chosen} was not executed, the second neighborhood method N_2 extends the first method N_1 such that besides trying to fix the executability condition (iv), the remaining conditions (ii) and (iii) are also tried to be fixed to execute the suborder o_{chosen}. Thus, if the item requested is previously sold, then the corresponding suborder o_x of a possibly different participant which was executed before is undone, making the item available again. Then, the suborder o_{chosen} is executed without checking the remaining executability conditions.

Note that the actions of undoing the suborder o_x and executing the suborder o_{chosen}

Input: The initial solution Sol, the suborder o_{chosen} to be executed or undone.
Output: The neighbor solution Sol' or $neighborNotFound$.

Set $Sol' \leftarrow Sol$
if the suborder o_{chosen} is already executed **then**
  Undo the suborder o_{chosen}
  if the budget of the participant h_x who owns the item requested in the suborder $o_{chosen} < 0$ **then**
    {Improve the budget of the participant h_x if possible}
    Set $budgetInDeficit \leftarrow$executeOrders($Sol', h_x, o_{excluded} = o_{chosen}$)
    if $budgetInDeficit$ **then**
      return $neighborNotFound$ {Unable to close the budget deficit}
    end if
  end if
  return the neighbor solution Sol'
else
  if the item t_x requested in suborder o_{chosen} is already sold in the suborder o_x of a participant h_x **then**
    Undo the suborder o_x
  end if
  Execute the suborder o_{chosen}
  $limitExceeded \leftarrow$ ensureOrderLimit(Sol', o_{chosen})
  if $limitExceeded$ **then**
    return $neighborNotFound$ {Unable to set the suborder limit of the order}
  end if
  if the budget of the participant h_x who posted the suborder $o_{chosen} < 0$ **then**
    Set $budgetInDeficit \leftarrow$executeOrders($Sol', h_x, o_{excluded} = NULL$)
    if $budgetInDeficit$ **then**
      Set $budgetInDeficit \leftarrow$undoOrders($Sol', h_x, o_{excluded} = o_{chosen}$)
    end if
    if $budgetInDeficit$ **then**
      return $neighborNotFound$ {Unable to close the budget deficit}
    end if
  end if
  return the neighbor solution Sol'
end if

Algorithm 3: Pseudocode for Neighborhood Method N_2

do not introduce a budget deficit for the owner of the item since her item is sold again to another participant who posted the suborder o_{chosen}. However, it may cause a budget deficit for the participant who posted the suborder o_{chosen} which is handled in the last part of the method.

The second issue with these actions is that after executing the suborder o_{chosen}, the limit of the purchase order O_x which includes the suborder o_{chosen} may have been exceeded such that two items may have been purchased in this purchase order. If this is the case, then the method finds the other suborder o_x which was already executed in the same purchase order. After that, the method checks whether the budget of the participant who sells the item requested in the suborder o_x becomes deficit or not. If not, the suborder o_x is undone, and the limit issue is resolved. Otherwise, the neighborhood method N_2 returns $neighborNotFound$. These steps of the neighborhood method N_2 are summarized as *ensureOrderLimit* method in Algorithm 4.

```
Input: The solution Sol′, the most recently executed suborder o_chosen
Output: The boolean flag limitExceeded indicating whether the order limit constraint is
    violated or not
Set O_x ← the order which comprises o_chosen
Set limitExceeded ← false
if the order O_x have two suborders that are already executed then
    Set limitExceeded ← true
    Find the other suborder o_x different than o_chosen which is already executed in the order O_x
    Set h_y ← the participant who sells the item requested in o_x
    if the budget of the participant h_y ≥ the price of o_x then
        Undo the suborder o_x
        Set limitExceeded ← false
    end if
end if
return limitExceeded
```

Algorithm 4: Pseudocode for ensureOrderLimit Method

The third and the final issue to be handled is the possible budget deficit of the participant h_x who posts the suborder o_{chosen}. If the budget of the participant h_x is in deficit, as in the first method N_1, the neighborhood method N_2 tries to execute the suborders which request an item of the participant h_x to close the deficit. However, if the executed suborders are not enough to close the deficit, unlike the neighborhood method N_1, N_2 tries to undo the already executed suborders posted by the participant h_x as shown in Algorithm 5 as *undoOrders* method. Undoing a suborder is only allowed if this action does not cause a budget deficit for the participant who sells the item requested in this suborder. Note that unlike the *executeOrders* method, *undoOrders* method executes the suborders in the ascending order of weight values, that is the method first tries to undo the order with the lowest weight value. Also, the method stops undoing the suborders when the deficit is closed. If all of these issues are resolved, i.e. the executability conditions (ii)-(iv) are satisfied, the neighborhood method returns the neighbor solution Sol'. Otherwise, $neighborNotFound$ is returned.

Input: The solution Sol', the participant h_x whose budget is in deficit, the suborder $o_{excluded}$ to be excluded
Output: The boolean flag *budgetInDeficit* indicating whether the budget of the participant h_x remains in deficit or not

Set $budgetInDeficit$ ←**false**
if the budget of the participant $h_x < 0$ **then**
  {Improve the budget of the participant h_x by undoing her already executed purchase orders}
  Set $budgetInDeficit$ ←**true**
  Build the list L_x of suborders of the participant h_x
  Sort L_x according to weight values w of suborders in *ascending* order
  for all suborder o_x in L_x **do**
    if $o_x = o_{excluded}$ **then**
      continue
    end if
    if the suborder o_x is already executed **and** the budget of the participant h_x who owns the item requested in the suborder $o_x \geq$ the price of o_x **and** $budgetInDeficit$ **then**
      Undo the suborder o_x
      if the budget of the participant $h_x \geq 0$ **then**
        Set $budgetInDeficit$ ←FALSE
      end if
    end if
  end for
end if
return $budgetInDeficit$

Algorithm 5: Pseudocode for undoOrders Method

As a summary, considering the bit string representation of solutions, the presented neighborhood methods return a neighbor solution whose *at least* one bit is flipped compared to the original solution. If flipping one bit makes the solution infeasible then both methods take different measures to satisfy feasibility by flipping a set of other bits. Note that the neighborhood method N_2 is more complex than N_1 causing possibly larger neighborhood size for a given solution, however at the same time requiring more CPU power.

6.2 Heuristic Methods for the MCP

In this study Hill Climbing, Iterated Local Search, Ant Colony Optimization, Artificial Bee Colony and Genetic Algorithm based heuristic methods are proposed for the MCP. Since they are well-known methods in the literature, they will be introduced briefly focusing on the parameters and the necessary implementation details. Note that the software package prepared for this study consists of more than 10,000 lines of code.

Hill Climbing (HC-best): This method starts with a given feasible solution (note that all zero solution is also feasible) and checks its feasible neighbor solutions using one of the proposed neighbor methods N_1 or N_2. If the best solution among the neighbor solutions is better than the current solution, then the best solution is accepted as the (new) current solution. Next, the method explores the (new) neighborhood of the current solution for a better solution. The method continues in this manner until the current solution is better than all the solutions in the defined neighborhood of the current solution. The method returns this final current solution as the best solution found. The pseudocode for this method is presented in Algorithm 6.

Input: The initial solution *initialSol*, the neighborhood method $N(x)$
Output: The best solution found *currentSol*.

```
Set currentSol ← initialSol
Set isImproved ← true
while isImproved = true do
   Set isImproved ← false
   Set bestSol ← currentSol
   for all suborder o_x in currentSol do
      Set neighSol ← N(currentSol, o_x)
      if neighSol = neighborNotFound then
         continue {forall loop}
      end if
      if neighSol is better than bestSol then
         Set bestSol ← neighSol
      end if
   end for
   if bestSol is better than currentSol then
      Set currentSol ← bestSol
      Set isImproved ← true
   end if
end while
return currentSol {the best solution found}
```

Algorithm 6: Pseudocode for Hill Climbing Method

Stochastic Hill Climbing (HC-random): In the stochastic variant of the hill climbing method, instead of a deterministic search, a stochastic search is conducted in the solution space. Thus, instead of finding the best neighbor solution to the current solution, the method evaluates neighbor solutions which are determined using the tournament selection method. In this selection method, a tournament vector of neighbor solutions is prepared. If the best solution in this vector is better than the current solution, it is accepted as the (new) current solution, otherwise, the current solution remains the same. Then, another neighbor solution to the current solution is generated by generating a new tournament vector. The method terminates if the current solution does not improve after a predefined number of iterations which is equal to the number of suborders in the given instance.

Iterated Local Search (ILS-best): Although runs fast, the hill climbing method has a major drawback of getting stuck at the local optima. To avoid this to a degree, the hill climbing method can be executed many times with different random initial solutions which is called the *hill climbing with random restarts* method. In this case, the method makes a random search in the space of local optima. Iterated local search method searches this space of local optima intelligently [31].

The pseudocode for the iterated local search method as adapted to the MCP is presented in Algorithm 7. Given an initial solution, the method first finds a local optimum using the hill climbing method. If this local optimum is better than the best local optimum found so far, then the best local optimum is updated. After finding the local optimum, the method finds a new initial solution by undoing some suborders of the found local optimum which are selected by using an adaptive probabilistic mechanism. In this mechanism, for each already executed suborder of the local optimum, a random number, called *dice*, is generated based on the continuous uniform distribution in the range [0,1). If *dice* is less than or equal to the adaptive undoing probability value $adaptiveProbVal$, then the corresponding suborder is *tried* to be undone. As noted in the neighborhood methods presented, undoing a suborder may cause a budget deficit for the owner of the item sold, therefore, the ILS-best method also tries to close this deficit by calling $executeOrders$ method.

If the deficit cannot be closed, then the method skips to the next already executed suborder in the found local optimum.

The adaptive undoing probability value $adaptiveProbVal$ is calculated as:

$$adaptiveProbVal = undoProbVal * \left(1 - \frac{\text{weight value of } o_x}{\text{max weight value in all suborders}}\right) \quad (18)$$

where $undoProbVal = 0.5$ is a parameter of the ILS-best method. Note that $adaptiveProbVal$ is close to the value of $undoProbVal$ for the suborders with smaller weight values and close to the zero for the suborders with higher weight values. Therefore, the suborders to be undone are selected using inverse weight proportionate selection.

Iterated Local Search with Stochastic Hill Climbing (ILS-random): This method is the same as the ILS-best method with an exception of using the stochastic hill climbing (HC-random) method instead of the hill climbing (HC-best) method for finding the local optima of a given solution.

Ant Colony Optimization (ACO): The previous methods are improvement heuristic methods that inputs an initial solution and try to improve this solution. However, the Ant Colony Optimization (ACO) method is a swarm intelligence method that works on a population of solutions.

The ACO method was first proposed by Dorigo et al. [32, 33] in the 1990s which is inspired by the observations on the real ant colonies. It tries to mimic the ants' search for food, more specifically, ants' method for finding the shortest path between a food source and their nests. Initially, ants leave their nests and arbitrarily walk around looking for a food source. During this search, they leave a substance called trail pheromone on the ground marking their path. Ants can also smell the pheromone. An ant coming out of its nest follows a path which it chooses probabilistically based on the pheromone levels and reinforces the pheromone trail by leaving its pheromone on the chosen path.

In this study, Ant Colony System [34, 35] based ACO method was used because of its efficiency. The pseudo code of this method is presented in Algorithm 8.

Input: The initial solution *initialSol*, the neighborhood method $N(x)$, the number of maximum local search iterations to be made if the solution does not improve $returnAfterIter$, the probability value for undoing a suborder $undoProbVal$

Output: The best solution found *bestLocalOptSol*.

Set *currentSol* ← *initialSol*
Set *bestLocalOptSol* ← *initialSol*
Set $iterationNo \leftarrow 0$
while $iterationNo < returnAfterIter$ **do**
  Set *localOptSol* ←Hill_Climb(*currentSol*, $N(x)$)
  if *localOptSol* is better than *bestLocalOptSol* **then**
    Set *bestLocalOptSol* ← *localOptSol*
    Set $iterationNo \leftarrow 0$
  end if
  Set $newInitialSol$ ← *localOptSol*
  for all suborder o_x in $newInitialSol$ **do**
    Set $adaptiveProbVal \leftarrow undoProbVal *$ (1−weight of o_x / maximum weight in all suborders);
    Set *dice* ← a pseudorandom number from continous uniform distribution [0,1)
    if the suborder o_x is already executed **and** *dice* $\leq adaptiveProbVal$ **then**
      Undo the suborder o_x
      if the budget of the participant h_x who owns the item requested in the suborder $o_x < 0$ **then**
        {Improve the budget of the participant h_x if possible}
        Set $budgetInDeficit$ ←executeOrders($newInitialSol, h_x, o_{excluded} = o_x$)
        if $budgetInDeficit$ **then**
          Undo all the changes made to $newInitialSol$ in this iteration {the budget deficit is not closed, restore the previous feasible state}
        end if
      end if
    end if
  end for
  Set *currentSol* ← $newInitialSol$ {the budget deficit is closed, newInitialSol is feasible}
  Set $iterationNo \leftarrow iterationNo + 1$
end while
return *bestLocalOptSol* {the best local optimum solution found}

Algorithm 7: Pseudocode for Iterated Local Search Method

Input: The list of suborders $O \leftarrow \{o_1, o_2, \cdots, o_f\}$ of the purchase orders, the number of ants A, the neighborhood method $N(x)$, the maximum number of generations to stop if the solution does not improve $returnAfterIter$, the pheromone elitist selection rate q.
Output: The best solution found S_{best}.

$S_{best} \leftarrow \{\}$
Set $S_{initial} \leftarrow$Hill_Climb($\{\}$, $N(x)$)
$\gamma \leftarrow$ TotalWeight($S_{initial}$) / 10
Initialize the pheromone values of all suborders $\overrightarrow{p} = \langle p_1, p_2, \ldots, p_f \rangle$ to γ
Set $iterCount \leftarrow 0$
while $iterCount < returnAfterIter$ **do**
    for $i \leftarrow 1$ **to** A **do**
        {Construct solutions}
        Set $bestSolFound \leftarrow$ **false**
        Set $S_i \leftarrow \{\}$
        Set $O_{executable} \leftarrow$ list of executable suborders in S_i determined using $N(x)$
        while $O_{executable}$ is not empty **do**
            Select o_{chosen} among the suborders $O_{executable}$ using the Elitist Component Selection method with the probability q.
            Execute the suborder o_{chosen} in S_i
            {As a result of execution of the suborder o_{chosen}, a new set of executable suborders may exist becuase of the budget transfer.}
            Set $O_{executable} \leftarrow$ list of executable suborders in S_i determined using $N(x)$
        end while
        if TotalWeight(S_i) $>$ TotalWeight(S_{best}) **then**
            Set $S_{best} \leftarrow S_i$
            Set $bestSolFound \leftarrow$ **true**
        end if
    end for
    for all $p_i \in \overrightarrow{p}$ **do**
        {Pheromone Evaporation}
        Set $p_i \leftarrow 0.5\, p_i + 0.5\, \gamma$
        if Suborder o_i is executed in S_{best} **then**
            {Pheromone Update}
            Set $p_i \leftarrow 0.5 p_i + 0.5$ TotalWeight(S_{best})
        end if
    end for
    if $bestSolFound$ **then**
        Set $iterCount \leftarrow 0$
    else
        Set $iterCount \leftarrow iterCount + 1$
    end if
end while
return S_{best} {the best solution found}

Algorithm 8: Pseudocode for Ant Colony Optimization (Ant Colony System) Method

In this method, the initial pheromone value, γ, is calculated first. This value is found by scaling the total weight value of a local optimum solution which is found using the Hill Climb algorithm. The suborders, O, constitute the components of solutions. Therefore, a pheromone value, p_i, is used for each suborder in the problem instance. All the pheromone values constitute the pheromone vector, $\overrightarrow{p}$. Each component of the pheromone vector, $\overrightarrow{p}$, is initialized to the calculated initial pheromone value, γ.

After the pheromone vector, $\overrightarrow{p}$, is initialized, a set of new solutions is generated from scratch by using the Elitist Component Selection method [31] which represents the walk of an ant in the colony. Therefore, the number of solutions to be generated is same as the number of ants, A, which is a parameter of the ACO method. The best solution found during this phase is stored as S_{best}. After that, the pheromone evaporation step is carried out. In this step, the pheromone value, p_i, for each component, i.e. suborder o_i, is reduced using the following formula:

$$p_i = 0.5\ p_i + 0.5\ \gamma. \tag{19}$$

Then, the pheromone elitist update step is carried out. In this step, for each executed suborder in the best solution found s_{best}, the corresponding pheromone value p_i is incremented in accordance with the total weight value of the best solution found S_{best} using the following formula:

$$p_i = 0.5\ p_i + 0.5\ \text{TotalWeight}(S_{best}). \tag{20}$$

After the pheromone update step is completed, a new set of ants exit from their nest looking for new food sources, that is, a new set of A solutions are generated. The process stops when there is no improvement in the best solution found after 50 generations.

In this method, the Elitist Component Selection method [31], as mentioned above, is used when generating solutions. In this method, a desirability value is calculated for each suborder o_i using the following formula:

$$\text{Desirability}(o_i) = p_i \text{ Weight}(o_i)^2 \tag{21}$$

where p_i is the pheromone value of the suborder o_i. Then, a solution is generated from scratch by executing executable suborders one by one. The suborder to be executed is determined as follows:

- with probability $q = 0.9$ (which is called the pheromone elitist selection rate), among the executable suborders, the one with the highest desirability value is selected;
- otherwise, the suborder to be executed is determined using tournament selection method among the executable suborders.

This process continues until there is no executable suborder is left in the solution and this guarantees that all the generated solutions reside at the border of the feasible region. Note that due to the budget constraints, executing a suborder may cause previously inexecutable suborders to become executable, therefore after executing a suborder, the list of executable suborders should be updated.

Artificial Bee Colony (ABC): Introduced in 2005, Artificial Bee Colony (ABC) method [36, 37] is another swarm intelligence method which is inspired by the behavior of bee colonies. An artificial bee colony consists of employed bees, onlookers and scouts. Firstly, each employed bee is assigned to a food source and it looks for a new food source which is in close vicinity to the assigned food source. If the employed bee finds a new source, then it compares its nectar amount to the nectar amount of the assigned food source. If the former is higher, then the employed bee forgets the position of the assigned source and keeps the position of the new one in its memory. Otherwise, it keeps the position of the assigned source. When all employed bees finish searching, they return to their hive and inform the onlookers about the positions and the nectar amounts of the food sources in their memory. Each onlooker, then, selects one of the food sources considering the nectar amounts of the sources such that a source with higher amount of nectar has a higher probability to be selected.

Then, the onlookers look for another food sources which are in close vicinity of the selected food sources. Again, they memorize the locations of the new sources if they contain higher nectar amount than the selected sources. If a new source cannot be found in vicinity of a food source after a number of trials, that food source assumed to be consumed, and thus abandoned. Then the employed bee assigned to that source is converted to a scout. The scout then looks for a new arbitrary food source and when it is found, the abandoned source is replaced with the newly found source. Note that the positions of the food sources correspond to the solutions in the ABC method and the nectar amounts correspond to the fitness values of the solutions.

The pseudocode of the ABC method is given in [38]. However, the original ABC method was defined for continuous optimization. Therefore, its extension developed by [39] for combinatorial optimization problems is used in this study. In the ABC method for the MCP, initially, a population of SN random solutions is generated where SN denotes the number of food sources in the ABC context. Then, new solutions based on the existing solutions in the population are generated using Eq. (4) in [39] together with one of the introduced neighborhood methods N_1 and N_2. This simulates the task of the employed bees. If the neighbor solution has a higher objective (fitness) value than the current solution, then the current solution is replaced with the new solution in the population. Otherwise, the neighbor solution is discarded, and the current solution is kept in the population. Following the explanation of the inventor of the ABC algorithm in [40], for simulating the task of each onlooker, a solution is selected using the roulette wheel selection method, and then a new solution is generated again using Eq. (4) in [39] together with one of the introduced neighborhood methods N_1 and N_2. If a solution has not been replaced after $limit = 20$ (a predefined parameter) number of trials, then it is replaced with a random solution in the population and its trial count is reset. This simulates the task of scouts. After these steps, a new population of solutions is obtained, and the best solution found so far is saved. This whole population generation process is repeated until no improvement is observed in the best solution found in the last 50 generations.

Genetic Algorithm (GA): The genetic algorithm is a well-known population-based algorithm that tries to improve a population of solutions.

The genetic algorithm requires an encoding scheme for solutions, and the binary string representation of solutions is used for this purpose where each bit in the string identifies whether the corresponding suborder is executed or not. The initial population is populated using randomly generated feasible solutions. To generate a feasible random solution, first, a feasible solution is obtained using the hill climbing method and then randomly chosen suborders are undone while preserving feasibility.

After that, all the solutions in the population are evaluated and their fitness values are calculated. To generate the next population, a mating pool of parent solutions is selected using the binary tournament selection method. Then, for each pair of parent solutions, a uniform crossover operation with a probability of $c_p = 0.5$ is applied to obtain two child solutions while preventing possible positional bias. The pseudocode of the uniform crossover operator which is modified for the MCP using the introduced problem-specific neighborhood methods can be seen in Algorithm 9.

Input: Two parent solutions $parentSol1$ and $parentSol2$, the crossover probability c_p, the neighborhood method $N(x)$
Output: Two child solutions $childSol1$ and $childSol2$.
Set $childSol1 \leftarrow parentSol1$
Set $childSol2 \leftarrow parentSol2$
for all suborder o_{x1} in $childSol1$ **do**
  Set o_{x2} as the corresponding suborder in $childSol2$
  Set $dice \leftarrow$ a pseudorandom number from continous uniform distribution [0,1)
  if $(dice \leq c_p)$ **and** ((o_{x1} is executed **and** o_{x2} is **not** executed) **or** (o_{x1} is **not** executed **and** o_{x2} is executed)) **then**
    Set $childSol1 \leftarrow N(childSol1, o_{x1})$
    Set $childSol2 \leftarrow N(childSol2, o_{x2})$
  end if
end for
return the child solutions $childSol1$ and $childSol2$

Algorithm 9: Pseudocode for Crossover Operator (Uniform Crossover)

After the child solutions are obtained, each child solution is mutated using the bitwise mutation operator with a probability of $m_p = 0.1$. The pseudocode of the bitwise mutation operator which is modified for the MCP using the introduced problem-specific neighborhood methods can be seen in Algorithm 10.

Note that both the unmodified uniform crossover and mutation operators may cause infeasible solutions due to the structure of the MCP. Therefore, the neighborhood methods proposed in this work is also used in both operators to preserve the feasibility of child solutions.

After the next generation of solutions is obtained, the method evaluates the solutions in the next generation and keeps track of the best solution found. This whole population generation process is repeated until no improvement is observed in the best solution found in the last 50 generations.

Input: The solution to be mutated Sol, the bitwise mutation probability m_p, the neighborhood method $N(x)$
Output: The mutated solution $mutatedSol$.
Set $mutatedSol \leftarrow Sol$
for all suborder o_x in $mutatedSol$ **do**
  Set $dice \leftarrow$ a pseudorandom number from continous uniform distribution [0,1)
  if $dice \leq m_p$ **then**
    Set $mutatedSol \leftarrow N(mutatedSol, o_x)$
  end if
end for
return the mutated solution $mutatedSol$

Algorithm 10: Pseudocode for Mutation Operator (Bitwise Mutation)

7 Experimental Results

To evaluate the outcome of the PPBM model and the performances of the proposed heuristic methods, a test case generator based on the GNU Scientific Library [41] for the PPBM model has been developed. Using the generator, a test package consisting of 1920 test instances has been prepared. In this suite, the number of sales order and the number of purchase orders are distributed with the Poisson distribution with mean values between 250 and 1250 for simulating different problem sizes, and the number of suborders inside the purchase orders is distributed with the Poisson distribution with mean values between 1 and 5. For determining the items requested in the purchase orders, two approaches have been considered. In the *uniform selection* approach, the items requested (i.e. the suborders) are uniformly selected among all the items. In the *popularity-based* approach, items are divided into different sized clusters such that each cluster represents a set of substitutable items. Then, a random *popularity* value in the range [0,1] is assigned for each item in the clusters where higher values indicate more popular items. To select items requested in a purchase order, first, a cluster is uniformly selected among all clusters, then the items requested in the purchase order are determined using the roulette wheel selection method among the items in the same cluster based on the popularity values of the items. Thus, in this second approach, the likelihood of selecting a popular item is higher than selecting an unpopular item in a purchase order whereas in the first approach both have an equal chance to be selected.

For determining the prices of the items, the statistical information presented in the work of Ghose et al. [42] which is based on Amazon.com marketplace sales data is used. Finally, to determine the maximum amount of money m_i that a participant h_i wants to spend in the market, the following procedure is applied. First, a budget ratio br_i is determined using the normal distribution with a mean value ranging between 10% and 80%. Then, the budget amount m_i for a participant h_i is calculated as:

$$m_i = (b_i^{max} \ - \ b_i^{min}) \cdot \ br_i \ + \ b_i^{min} \tag{22}$$

where b_i^{min} the minimum amount of money that the participant h_i needs to spend in the market to purchase the cheapest item in his purchase orders given that all of her sales orders are executed (0 if she does not need to spend any money), and b_i^{max} is the maximum amount of money she needs to spend in the market to purchase all the items she wants even if none of her sales orders are executed. Note that when the budget ratios br_i of all participants are small, they have to sell their items in the market in order to be able to purchase the items they want, whereas when these ratios are close to 100%, they can purchase the items without selling their items.

Note that the test case generator allows full factorial testing, that is, the effect of any parameter can be studied while the other parameters take different values. The test suite was solved using the industry-standard MIP solver, Gurobi Optimizer version 8 [43], on a system consisting of a total of 16 cores clocked at 3.10 GHz with 8 GB of memory per core using 64-bit Linux operating system. A time limit of 7 hours per problem instance was defined for the MIP solver. Even though small to medium-sized instances were generated (large instances with tens of thousands of orders are not generated intentionally) and a high time limit is set per instance, among the 1920 instances in the test suite, the MIP solver could find optimal solutions for 1564 instances. While reporting the results of the experiments, only these optimal solutions are used to prevent a possible bias.

7.1 Outcomes of the PPBM Model for Different Objectives and Comparison to the Simulated Outcome of the Current Market System

In this experiment, the outcomes of the PPBM model for different objectives are evaluated, and a comparison to the estimated outcome of the current market system is provided. For this purpose, all the test instances were solved to the optimality for each of the four objective functions introduced in Section 5. Then, each problem instance was simulated under the current market system conditions to estimate a possible outcome of the market instance if the PPBM model is not used.

For simulating a test instance, first, the list L of all suborders is constructed and shuffled. However, during shuffling the preferences of the participants are also taken into consideration so that in this list the highly preferred suborders of a participant comes before the less preferred suborders. Next, the initial budget of every participant h_i is set to the maximum amount of money she is willing to spend in the market, m_i. Then, since the current market system is based on the first-come-first-served scheme, each suborder in the list L is tried to be executed one by one. If the current suborder in the list L is executable, then it is marked as executed, the corresponding item is marked as sold and the price of the item is transferred from the budget of the buyer to the budget of the seller. If the current suborder is not executable, the next suborder in the list is processed. Since the budgets of the participants change over time, the list is traversed many times until no more executable suborders are left executable. Thus, this simulation has an *optimistic assumption* that the participants never give up trying to purchase the items they want. Each test instance is simulated 100 times each with a different order of the list L.

Table 1 summarizes the mean and the standard deviation values of the following four criteria used for evaluating the market outcome for each objective function: (i) the trading volume, (ii) the number of items traded, (iii) the number of satisfied sellers (i.e. the participants with a seller role) whose at least one item is sold, and (iv) the number of satisfied buyers (i.e. participants with a buyer role) who at least purchased one item in the market.

Table 1: Comparison of the market outcomes of the current market simulation results and the PPBM model for each of the four objective functions.

	Trading Volume			# Items Traded			# Satisfied Sellers			# Satisfied Buyers		
	mean	stdev	imprv	mean	stdev	imprv	mean	stdev	imprv	mean	stdev	imprv
Current Market Sim.	940322	428637	0.0%	441.9	175.2	0.0%	194.7	35.6	0.0%	186.1	35.8	0.0%
PPBM-Trading Vol.	1312164	500068	**39.5%**	523.5	190.9	**18.5%**	211.0	27.2	**8.4%**	210.9	29.0	**13.3%**
PPBM-Items Trd.	1177216	455096	**25.2%**	535.5	190.4	**21.2%**	213.3	25.6	**9.5%**	212.6	27.7	**14.2%**
PPBM-Pref. Sum	1195073	460812	**27.1%**	535.2	190.4	**21.1%**	214.7	25.3	**10.3%**	211.5	28.0	**13.6%**
PPBM-Blend. Obj.	1302570	498680	**38.5%**	533.4	190.9	**20.7%**	213.2	25.9	**9.5%**	211.2	28.1	**13.5%**

Based on these four criteria, PPBM-Trading Vol. objective provides the highest trading volume among the other objectives as expected, however, it falls short for the remaining criteria since it is focused on executing the suborders with the highest price. In the case of the PPBM-Items Trd. objective, naturally it provides the highest number of traded items in the market, however, the trading volume provided with this objective is fairly low compared to the PPBM-Trading Vol. objective. PPBM-Pref. Sum objective, on the other hand, aims to maximize the preference values of the participants declared for their items. It is similar to the PPBM-Items Trd. objective in the sense that it tries to maximize the weighted number of traded items. Therefore, the outcomes of both of these objectives are quite similar according to these criteria (except for the fairness criteria which is discussed below). Finally, the PPBM model using the blended objective, i.e. PPBM-Blend. Obj., provides improved and at the same time, balanced outcomes that are close to the best improvement rate obtained for each of the four criteria. It is seen that the outcome of the PPBM model improves the estimated outcome of the current market system in all four criteria.

As explained in Section 4, the second objective, Eq.(3), of the PPBM model aims to distribute the optimal primary objective value between the participants as evenly as possible to provide fairness between the participants. To evaluate the fairness of the allocation provided by the PPBM model, Jain's fairness index is used. Based on Jain's et. al original work [44], for a system allocating resources to m participants, such that the participant h_i receives an allocation x_i, Jain's fairness index is defined for the system as

$$JFI(x_1, x_2, \ldots, x_m) = \frac{[\sum_i^m x_i]^2}{\sum_i^m x_i^2} \tag{23}$$

Jain's fairness index measures the equality of the allocation $x_1, x_2, \ldots, x_m$ and returns values between $1/m$ and 1. When all users receive the same allocation, i.e. $x_1 = x_2 = \ldots = x_m$, the index returns 1 which means the allocation is totally fair. However, if one participant gets all the resources and the others receive none, then the index returns $1/m$.

Using the Jain's fairness index, the fairness of the allocations provided by the estimated outcome of the current market system and the outcome PPBM model are evaluated using six criteria which can be seen in Table 2. For the participants with seller roles, the total revenue, the sum of the sales preference values, and the number of items sold criteria are used, whereas, for the participants with buyer roles, total expenses, the sum of the purchase preference values and the number of items purchased criteria are used.

Table 2: Comparison of the fairness of the allocations provided by the current market system and the PPBM model for each of the four objective functions using Jain's fairness index.

	Jain's Fairness Indices for Sellers						Jain's Fairness Indices for Buyers					
	Revenue		Sales Pref.		Items Sold		Expenses		Purch. Pref.		Items Purch.	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev
Current Market Sim.	0.80	0.13	0.83	0.12	0.74	0.14	0.78	0.07	0.85	0.07	0.74	0.09
PPBM-Trading Vol.	0.91	0.09	0.89	0.09	0.89	0.09	0.93	0.04	0.92	0.05	0.94	0.04
PPBM-Items Trd.	0.88	0.10	0.90	0.09	0.90	0.09	0.89	0.04	0.94	0.05	0.95	0.04
PPBM-Pref. Sum	0.89	0.10	0.91	0.08	0.91	0.08	0.89	0.04	0.95	0.04	0.94	0.05
PPBM-Blend. Obj.	0.91	0.09	0.90	0.09	0.90	0.09	0.92	0.05	0.94	0.04	0.94	0.05

The range of mean values of the fairness indices for the estimated outcome of the market simulation ranges between 0.74 to 0.85 for these criteria even though the simulation has an optimistic assumption that the buyers never give up trying. However, for the PPBM model with any objective, the lowest mean value for the index is 0.88 which goes up to 0.95. The PPBM model using the blended objective, i.e. PPBM-Blend. Obj., provides again the balanced results where the mean values lie between 0.90 and 0.94. Therefore, it can be concluded that the PPBM model provides fairer allocations of items sold in the market compared to the current market system based on the simulation results.

7.2 Performances of the Proposed Heuristic Methods

To evaluate the performances of the heuristic methods, all the optimally solved instances in the test suite are also solved using the proposed seven heuristic methods, HC-best, HC-random, ILS-best, ILS-random, ABC, ACO and GA. For the population-based heuristics ABC, ACO and GA, three different population sizes[2], i.e. 20, 50 and 100 are tested which are referred to X-20, X-50, and X-100 where X is ABC or ACO or GA. Also, each heuristic method is tested using both neighborhood methods N_1 and N_2. Thus, a total of 26 heuristic configurations are tested for each of the optimally solved test instances. Note that, in this section, the performances of the heuristics are presented using the objective function PPBM-Blended Obj. since this objective is considered to provide the best overall outcomes among all four objective functions. The performances of the heuristics for the other objective functions are quite similar to the PPBM-Blended Obj., therefore, they are not included in this study to provide clarity.

The quality of the solutions found by the heuristic methods is evaluated by the success ratio measure which is defined as:

$$\text{Success ratio of a sol.} = \frac{\text{Obj. val. found by the heur. sol.}}{\text{Optimal objective value}} \cdot 100\% \tag{24}$$

The mean values and the box plot of the success ratios of the solutions found by the proposed heuristic methods can be seen in Figure 4. The mean differences of the success ratios of the solutions are also analyzed using two-way analysis of variance (ANOVA) test at an $\alpha = 0.05$ significance level using the independent variables: heuristic method and neighborhood method. It is seen that there is a statistically significant interaction between the heuristic method and the neighborhood method for the success ratio, $F(12, 40638) = 30.965, p \leq .0005, \text{partial } \eta^2 = .009$. Therefore, since there exists an interaction effect, simple main effects are analyzed and reported [45]. The pairwise comparison results which indicate whether the mean differences are significant or not can be seen in Table 3, Table 4 and Table 5. Note that in order to prevent repetitions in the text, the term "significant" will be used in the remaining text to indicate that the corresponding mean difference is statistically significant at the $\alpha = 0.05$ significance level.

The two hill climbing methods, HC-best and HC-random, provides almost the same results (failed to reject the null hypothesis) with mean success ratios of approximately 86.3% and 86.1% for the neighborhood method N_1. The mean success ratios increase to

[2]Population size refers to the number of food sources in the ABC method and the number of ants in the ACO method.

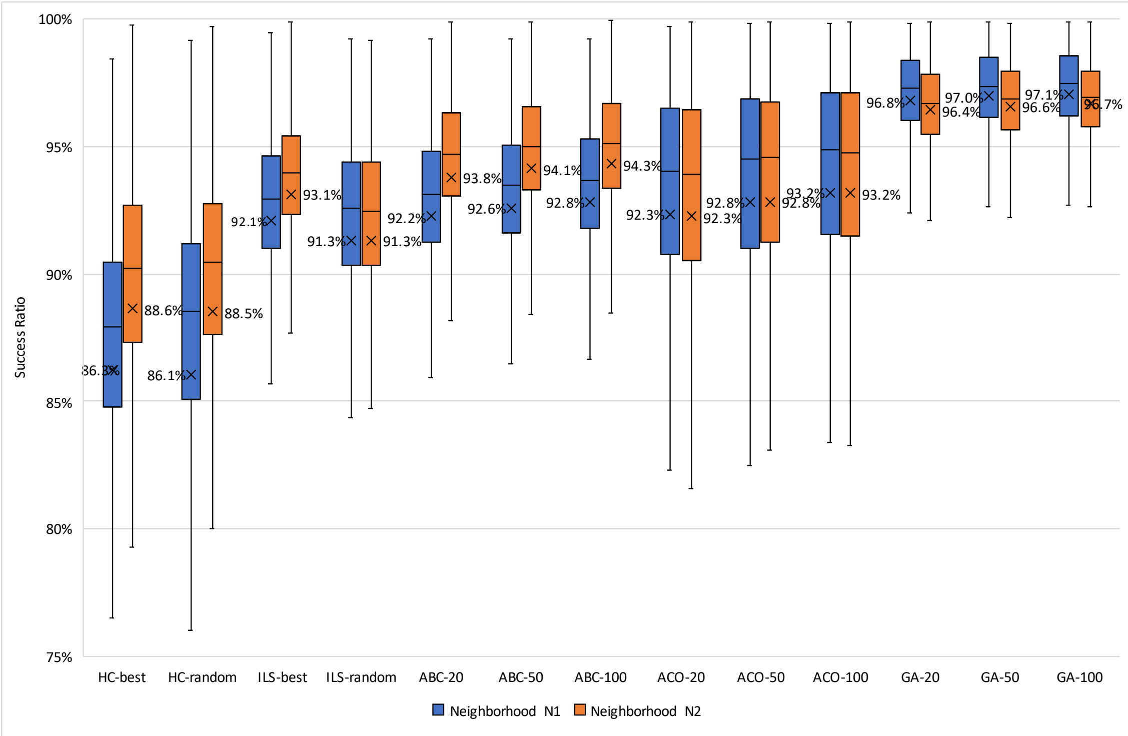

Figure 4: Boxplot presenting the success ratio of solutions of the heuristic methods for each of the neighborhood method N_1 and N_2. (x) indicates the mean success ratio of the corresponding heuristic method.

Table 3: Pairwise comparsions of the heuristic methods using the neighborhood method N_1 provided by the simple main effect analysis of the two-way ANOVA test. Each cell at the coordinate (X, Y) represents the difference between the mean success rate of the heuristic method at the row X and the mean success rate of the heuristic method at the column Y. (*) indicates that the corresponding mean difference is significant at the $\alpha = 0.05$ significance level.

Neigh. N_1	HC-best	HC-random	ILS-best	ILS-random	ABC-20	ABC-50	ABC-100	ACO-20	ACO-50	ACO-100	GA-20	GA-50	GA-100
HC-best	0	0.19	-5.83*	-5.05*	-5.99*	-6.33*	-6.58*	-6.07*	-6.56*	-6.95*	-10.53*	-10.70*	-10.81*
HC-random	-0.19	0	-6.02*	-5.24*	-6.18*	-6.52*	-6.77*	-6.26*	-6.75*	-7.14*	-10.72*	-10.89*	-11.00*
ILS-best	5.83*	6.02*	0	0.78*	-0.16	-0.50	-0.75*	-0.24	-0.73*	-1.12*	-4.70*	-4.87*	-4.98*
ILS-random	5.05*	5.24*	-0.78*	0	-0.94*	-1.28*	-1.53*	-1.03*	-1.52*	-1.91*	-5.48*	-5.65*	-5.77*
ABC-20	5.99*	6.18*	0.16	0.94*	0	-0.34	-0.59	-0.08	-0.57	-0.96*	-4.54*	-4.71*	-4.82*
ABC-50	6.33*	6.52*	0.50	1.28*	0.34	0	-0.25	0.26	-0.23	-0.62	-4.12*	-4.37*	-4.48*
ABC-100	6.58*	6.77*	0.75*	1.53*	0.59	0.25	0	0.50	0.01	-0.38	-3.95*	-4.12*	-4.24*
ACO-20	6.07*	6.26*	0.24	1.03*	0.08	-0.26	-0.50	0	-0.49	-0.88*	-4.46*	-4.63*	-4.74*
ACO-50	6.56*	6.75*	0.73*	1.52*	0.57	0.23	-0.01	0.49	0	-0.39	-3.97*	-4.14*	-4.25*
ACO-100	6.95*	7.14*	1.12*	1.91*	0.96*	0.62	0.38	0.88*	0.39	0	-3.58*	-3.75*	-3.86*
GA-20	10.53*	10.72*	4.70*	5.48*	4.54*	4.20*	3.95*	4.46*	3.97*	3.58*	0	-0.17	-0.28
GA-50	10.70*	10.89*	4.87*	5.65*	4.71*	4.37*	4.12*	4.63*	4.14*	3.75*	0.17	0	-0.11
GA-100	10.81*	11.00*	4.98*	5.77*	4.82*	4.48*	4.24*	4.74*	4.25*	3.86*	0.28	0.11	0

Table 4: Pairwise comparsions of the heuristic methods using the neighborhood method N_2 provided by the simple main effect analysis of the two-way ANOVA test. Each cell at the coordinate (X, Y) represents the difference between the mean success rate of the heuristic method at the row X and the mean success rate of the heuristic method at the column Y. (*) indicates that the corresponding mean difference is significant at the $\alpha = 0.05$ significance level.

Neigh. N_2	HC-best	HC-random	ILS-best	ILS-random	ABC-20	ABC-50	ABC-100	ACO-20	ACO-50	ACO-100	GA-20	GA-50	GA-100
HC-best	0	0.12	-4.50*	-2.70*	-5.18*	-5.51*	-5.67*	-3.63*	-4.19*	-4.54*	-7.79*	-7.96*	-8.07*
HC-random	-0.12	0	-4.62*	-2.82*	-5.30*	-5.63*	-5.78*	-3.75*	-4.30*	-4.66*	-7.91*	-8.08*	-8.19*
ILS-best	4.50*	4.62*	0	1.80*	-0.68*	-1.01*	-1.17*	0.87*	0.32	-0.04	-3.29*	-3.46*	-3.57*
ILS-random	2.70*	2.82*	-1.80*	0	-2.48*	-2.81*	-2.97*	-0.93*	-1.48*	-1.84*	-5.09*	-5.26*	-5.37*
ABC-20	5.18*	5.30*	0.68*	2.48*	0	-0.33	-0.49	1.55*	1.00*	0.64	-2.61*	-2.78*	-2.89*
ABC-50	5.51*	5.63*	1.01*	2.81*	0.33	0	-0.16	1.87*	1.32*	0.97*	-2.29*	-2.45*	-2.57*
ABC-100	5.67*	5.78*	1.17*	2.97*	0.49	0.16	0	2.03*	1.48*	1.12*	-2.13*	-2.29*	-2.41*
ACO-20	3.63*	3.75*	-0.87*	0.93*	-1.55*	-1.87*	-2.03*	0	-0.55	-0.91*	-4.16*	-4.33*	-4.44*
ACO-50	4.19*	4.30*	-0.32	1.48*	-1.00*	-1.32*	-1.48*	0.55	0	-0.36	-3.61*	-3.77*	-3.89*
ACO-100	4.54*	4.66*	0.04	1.84*	-0.64	-0.97*	-1.12*	0.91*	0.36	0	-3.25*	-3.42*	-3.53*
GA-20	7.79*	7.91*	3.29*	5.09*	2.61*	2.29*	2.13*	4.16*	3.61*	3.25*	0	-0.17	-0.28
GA-50	7.96*	8.08*	3.46*	5.26*	2.78*	2.45*	2.29*	4.33*	3.77*	3.42*	0.17	0	-0.11
GA-100	8.07*	8.19*	3.57*	5.37*	2.89*	2.57*	2.41*	4.44*	3.89*	3.53*	0.28	0.11	0

approximately 88.6% and 88.5% when the neighborhood method N_2 is used. This increase is statistically significant.

The iterated local search methods provide better solutions than the hill climbing methods as expected since they search through the space of local optima. The mean success ratios for these methods are between 91.3% and 93.1% which are significantly higher than the mean success ratios of HC-best and HC-random for both neighborhood methods. Also different from the hill climbing methods, in the ILS method, best-first and stochastic approaches provide different results. ILS-best provides slightly but significantly better solutions compared to its stochastic counterpart. Finally, the choice of the neighborhood method is insignificant for the ILS-random method, whereas, for the ILS-best method, the neighborhood method N_2 provides significantly better results than the method N_1.

The ABC method, being a population-based method, provides success ratios between approximately 92.2% and 94.1% varying with the population size (the number of food sources to be exact) and the used neighborhood method. Since it works with a population of solutions rather than a single solution, the ABC method provide significantly better solutions than the ILS methods in general. The only exception is that the null hypothesis is failed to be rejected for the mean differences between the ABC and ILS-best for the neighborhood method N_1. Also, for the ABC method, the neighborhood method N_2

Table 5: Pairwise comparsions of the neighborhood methods N_1 and N_2 for each heuristic method provided by the simple main effect analysis of the two-way ANOVA test. Each cell represents the difference between the mean success rate of the heuristic method using the neighborhood method N_1 and the mean success rate of the heuristic method using the neighborhood method N_2. (*) indicates that the corresponding mean difference is significant at the $\alpha = 0.05$ significance level.

Heuristic	Difference in means between N_1 and N_2
HC-best	-2.373*
HC-random	-2.444*
ILS-best	-1.046*
ILS-random	-0.027
ABC-20	-1.563*
ABC-50	-1.549*
ABC-100	-1.463*
ACO-20	0.066
ACO-50	0.003
ACO-100	0.036
GA-20	0.364
GA-50	0.367
GA-100	0.368

provides approximately 1.5% better results in terms of success ratio which is also shown to be statistically significant.

The ACO method, on the other hand, provides success ratios between approximately 92.3% and 93.2%. This range is quite close to the success ratios the ABC method when the neighborhood N_1 is used. Therefore, the statistical analysis does not indicate a significant difference between the results of ACO and the ABC methods for the neighborhood method N_1 (the only exception is that the mean success ratio of ACO-100 is significantly higher than the mean success ratio of ABC-20). However, for the neighborhood method N_2, the ABC method provides significantly better results than the ACO method. The selection of neighborhood method is insignificant on the mean success ratio for the ACO method.

Finally, the genetic algorithm provides the best solutions among all heuristics with the mean success ratios in the range of 96.4% to 97.1%. It provides significantly better solutions than all the other heuristic methods presented for each of the neighborhood method N_1 and N_2 even when the population size is 20. The neighborhood method N_1 when used with the GA provides better solutions compared to the neighborhood method N_2, however, the mean difference cannot be shown to be statistically significant.

In all population-based heuristics, three population sizes, 20, 50 and 100, are tested. The mean success ratio increases slightly as the population size increases for all the methods; however, the statistical test is unable to show a significant difference among these values at the $\alpha = 0.05$ significance level in general (only the mean difference between ACO-20 and ACO-100 is significant). Therefore, it is considered that using a population size greater than 20 may not be needed for these heuristic methods.

The market outcomes provided by the heuristic methods can be seen in Table 6. The general trend of the heuristic performances for the objective value is also observed in the four criteria of the market outcome evaluation. The genetic algorithm provides the best market outcome where the mean trading volume is within 4% of the optimal solution, the mean number of items traded is within 3% and the mean numbers of satisfied sellers and the buyers are within only 1%.

Finally, using the Jain's fairness index, the fairness of the allocations provided by the heuristic methods are compared to the optimal allocation found by the MIP solver in Table 7. All the proposed methods provide solutions the mean index values of which are within 8% of the optimal allocation for all six criteria. Again, the genetic algorithm provides the fairest results among all the proposed heuristic methods only within 1% of the mean index value of the optimal solutions.

The comparison of the running times of the proposed heuristic methods and the MIP solver can be seen in Figure 5. Note that for all population-based heuristics, ABC, ACO and GA, the same population sizes and the same stopping criterion is used. All these methods terminate when the best solution cannot be improved in the last 50 generations.

The mean running time of the genetic algorithm using the neighborhood method N_1 increases almost linearly as the population size increases as expected in the range of 0.5 seconds to 3.4 seconds where the most difficult instance requires approximately 13 seconds when the population size is 100. These values slightly increase when the neighborhood method N_2 is used instead of N_1.

The ACO method is the slowest method among all the heuristic methods with mean running times in the range of 8.7 seconds to 43.5 seconds. The reason for this slow performance is that the ACO method constructs a population of solutions from scratch at every generation. The maximum running time for this method is 642 seconds for the most

Table 6: Comparison of the market outcomes of the heuristic methods and the optimal solutions found by the MIP solver.

		Trading Volume			# Items Traded			# Satisfied Sellers			# Satisfied Buyers		
Neigh.	Heuristic	mean	stdev	success	mean	stdev	success	mean	stdev	success	mean	stdev	success
MIP Solver (Opt.)		1302570	498680	100.0%	533.4	190.9	100.0%	213.2	25.9	100.0%	211.2	28.1	100.0%
N_1	HC-best	1144969	462017	**87.9%**	461.4	172.2	**86.5%**	199.2	33.8	**93.4%**	203.8	34.0	**96.5%**
	HC-random	1161237	480644	**89.1%**	457.2	177.8	**85.7%**	197.9	35.7	**92.8%**	199.3	36.0	**94.3%**
	ILS-best	1206156	472450	**92.6%**	494.0	178.6	**92.6%**	207.4	29.6	**97.3%**	207.7	30.7	**98.3%**
	ILS-random	1207806	480359	**92.7%**	490.5	180.7	**92.0%**	205.9	30.4	**96.6%**	203.3	31.5	**96.3%**
	ABC-20	1217809	475549	**93.5%**	494.1	180.7	**92.6%**	206.1	30.4	**96.7%**	204.7	31.2	**96.9%**
	ABC-50	1221552	475349	**93.8%**	495.6	180.7	**92.9%**	206.4	30.1	**96.8%**	204.9	31.0	**97.0%**
	ABC-100	1223962	476007	**94.0%**	497.0	181.1	**93.2%**	206.8	29.8	**97.0%**	205.1	30.7	**97.1%**
	ACO-20	1220349	487861	**93.7%**	494.0	184.3	**92.6%**	205.4	30.7	**96.4%**	203.8	31.8	**96.5%**
	ACO-50	1225802	489771	**94.1%**	496.6	185.1	**93.1%**	205.9	30.4	**96.6%**	204.1	31.6	**96.6%**
	ACO-100	1230123	490417	**94.4%**	498.8	185.4	**93.5%**	206.4	30.2	**96.8%**	204.5	31.4	**96.8%**
	GA-20	1255357	487052	**96.4%**	522.5	187.1	**98.0%**	211.9	27.0	**99.4%**	208.9	29.1	**98.9%**
	GA-50	1256951	486419	**96.5%**	523.0	187.0	**98.1%**	212.0	26.9	**99.4%**	208.9	29.0	**98.9%**
	GA-100	1258021	485996	**96.6%**	523.3	186.9	**98.1%**	212.1	26.6	**99.5%**	209.0	28.9	**98.9%**
N_2	HC-best	1175422	474817	**90.2%**	473.4	178.3	**88.8%**	201.5	33.4	**94.5%**	205.1	33.5	**97.1%**
	HC-random	1180191	482915	**90.6%**	471.8	179.4	**88.5%**	201.9	33.0	**94.7%**	203.0	34.1	**96.1%**
	ILS-best	1223934	479670	**94.0%**	497.2	180.8	**93.2%**	207.8	29.6	**97.5%**	208.3	30.8	**98.6%**
	ILS-random	1208012	479366	**92.7%**	490.3	180.3	**91.9%**	205.8	30.2	**96.5%**	203.3	31.5	**96.3%**
	ABC-20	1237390	484092	**95.0%**	500.4	183.4	**93.8%**	207.2	30.0	**97.2%**	206.7	31.1	**97.9%**
	ABC-50	1240663	484226	**95.2%**	501.9	183.6	**94.1%**	207.7	29.7	**97.4%**	206.9	30.9	**98.0%**
	ABC-100	1242403	484082	**95.4%**	502.7	183.6	**94.3%**	207.9	29.5	**97.5%**	207.0	30.8	**98.0%**
	ACO-20	1220047	488248	**93.7%**	493.4	184.5	**92.5%**	205.3	31.0	**96.3%**	203.8	31.9	**96.5%**
	ACO-50	1225943	489996	**94.1%**	496.5	185.4	**93.1%**	205.8	30.7	**96.5%**	204.1	31.7	**96.6%**
	ACO-100	1229816	490467	**94.4%**	498.3	185.7	**93.4%**	206.2	30.2	**96.7%**	204.5	31.4	**96.8%**
	GA-20	1254647	483564	**96.3%**	516.5	183.9	**96.8%**	211.6	26.9	**99.2%**	208.9	28.9	**98.9%**
	GA-50	1256459	483456	**96.5%**	517.1	183.7	**97.0%**	211.7	26.8	**99.3%**	209.0	28.8	**99.0%**
	GA-100	1257340	483123	**96.5%**	517.6	183.7	**97.0%**	211.8	26.8	**99.3%**	209.0	28.8	**99.0%**

Table 7: Comparison of the fairness of the allocations provided by the heuristic methods and the optimal solution found by the MIP solver using Jain's fairness index.

		Jain's Fairness Indices for Sellers						Jain's Fairness Indices for Buyers					
		Revenue		Sales Pref.		Items Sold		Expenses		Purch. Pref.		Items Purch.	
Neigh.	Heuristic	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev
MIP Solver (Opt.)		0.91	0.09	0.90	0.09	0.90	0.09	0.92	0.05	0.94	0.04	0.94	0.05
N_1	HC-best	0.83	0.12	0.83	0.12	0.83	0.12	0.85	0.06	0.88	0.06	0.88	0.06
	HC-random	0.83	0.13	0.82	0.12	0.82	0.12	0.84	0.07	0.86	0.07	0.86	0.07
	ILS-best	0.87	0.11	0.87	0.10	0.87	0.10	0.88	0.06	0.91	0.05	0.91	0.05
	ILS-random	0.87	0.11	0.86	0.10	0.86	0.10	0.86	0.06	0.89	0.06	0.89	0.06
	ABC-20	0.87	0.11	0.86	0.11	0.86	0.11	0.87	0.06	0.90	0.05	0.90	0.06
	ABC-50	0.87	0.11	0.86	0.11	0.86	0.10	0.87	0.06	0.90	0.05	0.90	0.06
	ABC-100	0.87	0.11	0.87	0.10	0.87	0.10	0.87	0.06	0.90	0.05	0.90	0.06
	ACO-20	0.87	0.12	0.86	0.11	0.86	0.11	0.87	0.06	0.89	0.06	0.90	0.06
	ACO-50	0.87	0.11	0.86	0.11	0.86	0.11	0.87	0.06	0.90	0.06	0.90	0.06
	ACO-100	0.87	0.11	0.87	0.11	0.87	0.11	0.87	0.06	0.90	0.06	0.90	0.06
	GA-20	0.90	0.10	0.89	0.09	0.89	0.09	0.89	0.05	0.93	0.05	0.93	0.05
	GA-50	0.90	0.10	0.89	0.09	0.89	0.09	0.89	0.05	0.93	0.05	0.93	0.05
	GA-100	0.90	0.09	0.89	0.09	0.89	0.09	0.90	0.05	0.93	0.05	0.93	0.05
N_2	HC-best	0.85	0.12	0.84	0.12	0.84	0.12	0.86	0.06	0.89	0.06	0.89	0.06
	HC-random	0.84	0.12	0.84	0.11	0.84	0.11	0.86	0.06	0.88	0.06	0.88	0.06
	ILS-best	0.88	0.11	0.87	0.10	0.87	0.10	0.89	0.05	0.91	0.05	0.91	0.05
	ILS-random	0.87	0.11	0.86	0.10	0.86	0.10	0.86	0.06	0.89	0.06	0.89	0.06
	ABC-20	0.88	0.11	0.87	0.11	0.87	0.11	0.88	0.06	0.91	0.05	0.91	0.05
	ABC-50	0.88	0.11	0.87	0.10	0.87	0.10	0.88	0.06	0.91	0.05	0.91	0.05
	ABC-100	0.88	0.11	0.87	0.10	0.87	0.10	0.88	0.05	0.91	0.05	0.91	0.05
	ACO-20	0.87	0.12	0.86	0.11	0.86	0.11	0.87	0.06	0.89	0.06	0.90	0.06
	ACO-50	0.87	0.12	0.86	0.11	0.86	0.11	0.87	0.06	0.90	0.06	0.90	0.06
	ACO-100	0.87	0.11	0.86	0.11	0.86	0.11	0.87	0.06	0.90	0.06	0.90	0.06
	GA-20	0.90	0.09	0.89	0.09	0.89	0.09	0.89	0.05	0.92	0.05	0.93	0.05
	GA-50	0.90	0.09	0.89	0.09	0.89	0.09	0.89	0.05	0.92	0.05	0.93	0.05
	GA-100	0.90	0.09	0.89	0.09	0.89	0.09	0.89	0.05	0.92	0.05	0.93	0.05

difficult instance.

The mean running times of the ABC method is in between 2.9 seconds and 17.8 seconds depending on the population size and the neighborhood function used. It is faster than the ACO method, since it does not construct solutions from scratch in the successive generations. However, it is also considerably slower than the GA. The maximum running time of the ABC method is observed as 88 seconds.

The running times of the ILS algorithm varies greatly depending on whether the best-first or stochastic approach is taken. The running time of the ILS-best method is comparable to the genetic algorithm with population size 100. However, ILS-random runs 10x to 20x faster than the ILS-best method on average depending on the used neighborhood method. Hill climbing algorithms are the fastest methods among all with a mean running time of less than 0.4 seconds, and a maximum running time of 4.1 seconds. The complexity of the neighborhood method N_2 over the neighborhood method N_1 can also be seen in the mean running time values. For all the heuristic methods, using the neighborhood method N_2 increases the mean running times of the corresponding method due to its greater complexity.

Although, large problem instances are intentionally avoided in the test suite in order to be able to find the optimal solutions for unbiased comparison, the mean running time of the MIP solver is approximately 150 seconds with a maximum value of approximately 6400 seconds. Note that the MIP solver was unable to find the optimal solutions for 356 cases in 7 hours (the time limit defined for each instance). These unsolvable instances are not included in the mean running time value of the MIP solver.

Note that the experimental results provided in this study are considered to be on par with the theoretical expectations for the proposed heuristics. The hill-climbing method is a local search heuristic that tries to improve a given initial solution by accepting only improving modifications to reach a local optimum with respect to the defined neighborhood relation. Therefore, the biggest drawback of this method is that it gets stuck at local optima. This is also observed experimentally that the success rate of hill-climbing methods is less than 90% on average. However, the stochastic hill-climbing method is the fastest method among the other proposed methods on average.

Different from the hill-climbing method, the iterated local search method does not

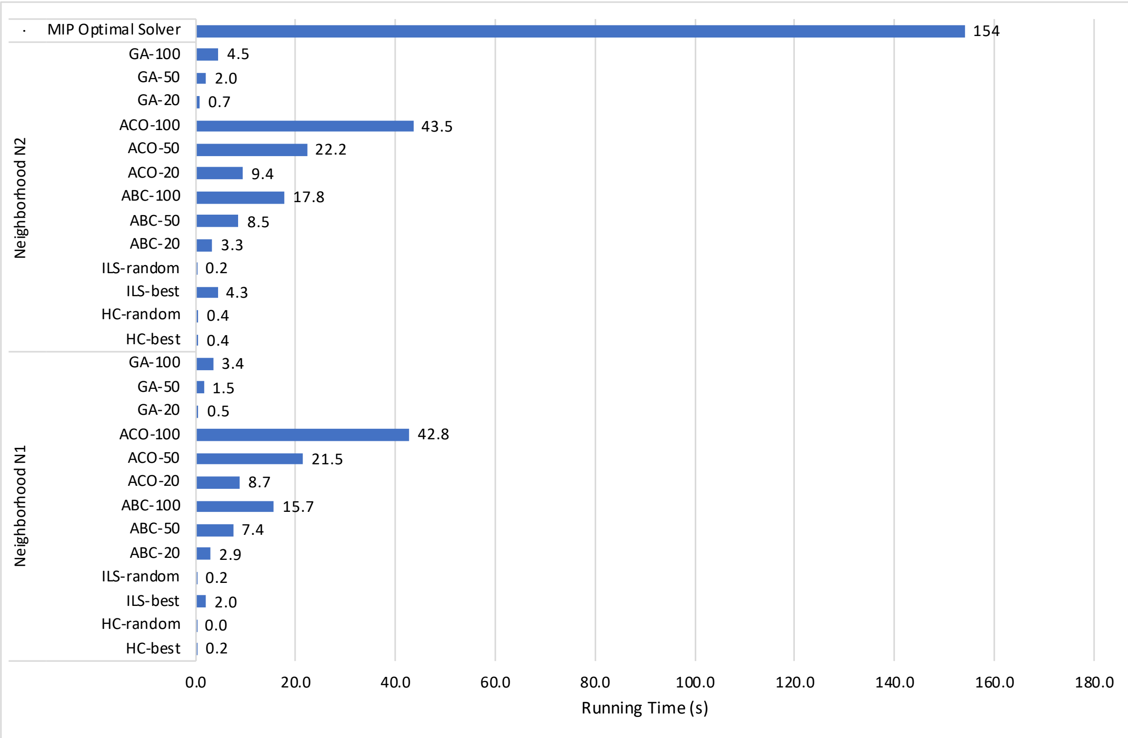

Figure 5: The mean running times (in seconds) of the MIP solver and the heuristic methods for each of the neighborhood method N_1 and N_2.

stop immediately when it finds a local optimum. Instead, it tries to escape from the local optimum by jumping to another solution in the solution space which is in close vicinity of the local optimum. Then, the method tries to improve this solution to reach possibly another local optimum. The iterated local search method uses the hill-climbing method for finding a local optimum. Therefore, it may be considered that the iterated local search method searches the local optima of the solution space, and hence, it is expected to provide better solutions than the hill-climbing method. This is also observed experimentally such that the average success ratio of the iterated local search methods is significantly higher than that of the hill-climbing methods reaching approximately 93%.

Both the hill-climbing and iterated local search methods work on a single solution to improve it. However, the artificial bee colony, ant colony optimization, and genetic algorithm are population-based methods that work on a population of solutions. Therefore, it is expected that they will come up with a better solution than the hill-climbing and iterated local search methods since they search more of the solution space. However, based on the experimental results, the gain to be obtained when using the artificial bee colony and ant colony optimization methods compared to the iterated local search is low, albeit significant. The genetic algorithm, on the other hand, is observed to search the solution space more efficiently than the other population-based methods. It provides higher quality solutions than the artificial bee colony and ant colony optimization methods also in less amount of time.

8 Conclusion

Compared to the traditional markets, electronic markets have the benefits of weakening time and space restrictions, reducing transaction costs, and allowing increased product variety [1]. Due to these benefits, in their seminal article, Malone et al. [46] envisioned that electronic markets would be the dominant institutions in the overall economic activity. Although there are many successful electronic markets for retail trade, as also stated by Clausen et al. [47], the full potential of resale trade remains to be captured.

Therefore, this study proposes an electronic market model, the PPBM model, for resale markets to increase the trading of used goods which has strong environmental and economic benefits in the circular economy. For this purpose, the PPBM model features several mechanisms such as (i) allowing the participants to spend the revenue to be obtained in the market for purchasing new items, (ii) preventing them from having a budget deficit, (iii) allowing them to express their substitutability preferences for the items they want to purchase, and (iv) allowing them to prioritize the items they want to sell and purchase.

In this study, the PPBM model and the associated market clearing problem, are formally defined. The market clearing problem is a combinatorial optimization problem that is formulated as a hierarchical multi-objective linear integer program with an adaptive objective function to provide fair allocation between the participants. Four different objective functions are introduced which can also be further extended depending on the intended market application. It is proven that the market clearing problem is intractable. Therefore, several heuristic methods are also proposed including hill climbing, iterated local search, artificial bee colony, ant colony optimization and genetic algorithms along with two problem-specific novel neighborhood methods. To estimate the performance of the model itself and the proposed heuristic methods, a large test suite consisting of approximately two thousand test instances is prepared.

The outcomes provided by the optimal solutions of the PPBM model using each introduced objective function are compared to the outcome of the current market system simulation under four different outcome criteria. It is observed that the proposed model provides a significant improvement in market allocation compared to the current market system. Additionally, the fairness of the allocations provided by the model is also evaluated using Jain's fairness criteria [44] the result of which indicates that the model provides more than 90% fair allocations under six different criteria defined for both sellers and buyers.

Next, the performances of the proposed heuristic methods are also compared to the optimal solutions found by an industry-standard general-purpose MIP solver. It is observed that the MIP solver is incapable of solving test instances even with approximately one thousand orders. However, the proposed heuristic methods execute much faster compared to the MIP solver, and the degradation in the quality of the solutions found is considered small especially for the genetic algorithm. The genetic algorithm using the problem-specific operators based on the proposed neighborhood methods provides the best solutions which are within 3% of the optimal solutions in terms of the objective value on average. Although this gap is higher for hill climbing and iterated local search methods, they can be preferred for solving large problem instances where it is not feasible to use the genetic algorithm due to its greater running time requirement.

Finally, the market outcomes provided by the heuristic methods are also compared to the optimal solutions of the model. As in the case of the objective values, the genetic algorithm again provides the best market outcomes for each of the defined criteria, the outcomes found by the genetic algorithm are at worst within 4% of the optimal solutions on average. Furthermore, the allocations found by this method are also at most 1% less fair compared to the optimal solutions.

As a conclusion, the empirical analysis suggests that the PPBM model will help to improve the market outcomes of the resale markets and contribute to the growth of circular economy while providing fair allocations between the participants. Although the optimization problem of the model is intractable, along with the proposed efficient heuristic methods, the model can be used in large scale resale markets with a small deviation from the optimal allocation. As a future work, further economic efficiency analyses can be conducted which would shed light on the expected performance of the model in real life.

Acknowledgments

This work has been supported by Marmara University Scientific Research Projects Coordination Unit under grant number FEN-A-130612-0218.

References

[1] Ali Haydar Özer. *Auction and Barter Models for Electronic Markets.* Phd Thesis, Bogazici University, Istanbul, 2011.

[2] Martin Grieger. Electronic marketplaces: A literature review and a call for supply chain management research. *European Journal of Operational Research*, 144(2):280–294, 2003. doi:10.1016/S0377-2217(02)00394-6.

[3] eBay. eBay Fast Facts. `https://investors.ebayinc.com/fast-facts/default.aspx`, 2020. (accessed 06.05.2020).

[4] Liran Einav, Chiara Farronato, Jonathan Levin, and Neel Sundaresan. Auctions versus Posted Prices in Online Markets. *Journal of Political Economy*, 126(1):178–215, 2018. doi:10.1086/695529.

[5] Alibaba. Alibaba Group Announces June Quarter 2019 Results. `https://www.alibabagroup.com/en/news/press_pdf/p190815.pdf`, 2019. (accessed 06.05.2020).

[6] Erik Brynjolfsson, Yu (Jeffrey) Hu, and Michael D. Smith. Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers. *Management Science*, 49(11):1580–1596, 2003. doi:10.1287/mnsc.49.11.1580.20580.

[7] J. Yannis Bakos. A Strategic Analysis of Electronic Marketplaces. *MIS Quarterly*, 15 (3):295–310, 1991. doi:10.2307/249641.

[8] J. Yannis Bakos. Reducing Buyer Search Costs: Implications for Electronic Marketplaces. *Management Science*, 43(12):1676–1692, 1997. doi:10.1287/mnsc.43.12.1676.

[9] Ho Geun Lee. Do electronic marketplaces lower the price of goods? *Communications of the ACM*, 41(1), 1998. doi:10.1145/268092.

[10] thredUP. 2019 Fashion Resale Market and Trend Report. `https://www.thredup.com/resale`, 2019. (accessed 06.05.2020).

[11] Sorilbran Stone. The Rise of Recommerce and Online Resale Sites. `https://www.theshelf.com/the-blog/resale-industry-recommerce-stats`, 2018. (accessed 06.05.2020).

[12] KD Market Insights. Global Pre-Owned Luxury Goods Market to Expand with Notable CAGR of 12.0% Between 2019 & 2024. `https://www.kdmarketinsights.com/pr/36/pre-owned-luxury-goods-market`, 2019. (accessed 06.05.2020).

[13] David Pearce. *Economics of Natural Resources and the Environment*. The Johns Hopkins University Press, Baltimore, 1989. ISBN 978-0-8018-3987-0.

[14] Patrizia Ghisellini, Catia Cialani, and Sergio Ulgiati. A review on circular economy: The expected transition to a balanced interplay of environmental and economic systems. *Journal of Cleaner Production*, 114:11–32, 2016. doi:10.1016/j.jclepro.2015.09.007.

[15] Brian Merchant. How many gallons of water does it take to make... `https://www.treehugger.com/clean-technology/how-many-gallons-of-water-does-it-take-to-make.html`, 2009. (accessed 06.05.2020).

[16] Oxfam. New shocking facts about the impact of fast fashion on our climate. `https://oxfamapps.org.uk/shop-blog/sustainable-fashion/new-shocking-facts-about-the-impact-of-fast-fashion-on-our-climate/`, 2019. (accessed 06.05.2020).

[17] Schibsted. Schibsted's Annual Report 2018. https://schibsted.com/news/schibsteds-annual-report-2018-has-been-published/, 2018. (accessed 06.05.2020).

[18] Patrick Bajari and Ali Hortaçsu. Economic Insights from Internet Auctions. *Journal of Economic Literature*, 42(2):457–486, 2004. doi:10.1257/0022051041409075.

[19] The Economist. In the great web bazaar. https://www.economist.com/special-report/2000/02/24/in-the-great-web-bazaar, 2000. (accessed 06.05.2020).

[20] Robert Zeithammer and Pengxuan Liu. When is auctioning preferred to posting a fixed selling price? Working Paper, University of Chicago, 2006.

[21] Robert G. Hammond. Comparing revenue from auctions and posted prices. *International Journal of Industrial Organization*, 28(1):1–9, 2010. doi:10.1016/j.ijindorg.2009.06.002.

[22] Robert G. Hammond. A structural model of competing sellers: Auctions and posted prices. *European Economic Review*, 60:52–68, 2013. doi:10.1016/j.euroecorev.2013.01.009.

[23] Christoph Bauner. Mechanism choice and the buy-it-now auction: A structural model of competing buyers and sellers. *International Journal of Industrial Organization*, 38:19–31, 2015. doi:10.1016/j.ijindorg.2014.12.001.

[24] Ali Haydar Özer. A double auction based mathematical market model and heuristics for internet-based secondhand durable good markets. *Computers & Operations Research*, 111:116–129, 2019. doi:10.1016/j.cor.2019.06.005.

[25] Kalyan Chatterjee and William Samuelson. Bargaining under Incomplete Information. *Operations Research*, 31(5):835–851, 1983. doi:10.1287/opre.31.5.835.

[26] Can Özturan. Network Flow Models for Electronic Barter Exchanges. *Journal of Organizational Computing and Electronic Commerce*, 14(3):175–194, 2004. doi:10.1207/s15327744joce1403_02.

[27] Ali Haydar Özer and Can Özturan. A direct barter model for course add/drop process. *Discrete Applied Mathematics*, 159(8):812–825, 2011. doi:10.1016/j.dam.2011.01.004.

[28] Ali Haydar Özer and Can Özturan. Multi-unit differential auction–barter model for electronic marketplaces. *Electronic Commerce Research and Applications*, 10(2):132–143, 2011. doi:10.1016/j.elerap.2010.03.002.

[29] William Stanley Jevons. *The Theory of Political Economy.* Macmillan and Co., London; New York, 1888.

[30] Michael M. Güntzer, Dieter Jungnickel, and Matthias Leclerc. Efficient algorithms for the clearing of interbank payments. *European Journal of Operational Research*, 106(1):212–219, 1998. doi:10.1016/S0377-2217(97)00265-8.

[31] Sean Luke. *Essentials of Metaheuristics.* Lulu, Morrisville, N.C., second edition, 2013. ISBN 978-1-300-54962-8.

[32] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Positive Feedback as a Search Strategy. Technical report, 1991.

[33] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, February 1996. ISSN 1941-0492. doi:10.1109/3477.484436.

[34] M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, April 1997. ISSN 1941-0026. doi:10.1109/4235.585892.

[35] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization.* MIT Press, MA, 2004.

[36] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, 2005.

[37] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, November 2007. ISSN 1573-2916. doi:10.1007/s10898-007-9149-x.

[38] Marjan Mernik, Shih-Hsi Liu, Dervis Karaboga, and Matej Črepinšek. On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. *Information Sciences*, 291:115–127, January 2015. ISSN 0020-0255. doi:10.1016/j.ins.2014.08.040.

[39] Dervis Karaboga and Bahriye Basturk. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In Patricia Melin, Oscar Castillo, Luis T. Aguilar, Janusz Kacprzyk, and Witold Pedrycz, editors, *Foundations of Fuzzy Logic and Soft Computing*, Lecture Notes in Computer Science, pages 789–798, Berlin, Heidelberg, 2007. Springer. ISBN 978-3-540-72950-1. doi:10.1007/978-3-540-72950-1_77.

[40] D. Karaboga and B. Basturk. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1):687–697, January 2008. ISSN 1568-4946. doi:10.1016/j.asoc.2007.05.007.

[41] GNU. GNU Scientific Library. `https://www.gnu.org/software/gsl/`, 2020. (accessed 06.05.2020).

[42] Anindya Ghose, Michael D. Smith, and Rahul Telang. Internet Exchanges for Used Books: An Empirical Analysis of Product Cannibalization and Welfare Impact. *Information Systems Research*, 17(1):3–19, 2006. doi:10.1287/isre.1050.0072.

[43] Gurobi. Gurobi Optimization. `https://www.gurobi.com/`, 2020. (accessed 06.05.2020).

[44] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. Technical Report TR-301, DEC, 1984.

[45] Scott E. Maxwell, Harold D. Delaney, and Ken Kelley. *Designing Experiments and Analyzing Data: A Model Comparison Perspective.* Psychology Press, New York, second edition, 2004.

[46] Thomas W. Malone, Joanne Yates, and Robert I. Benjamin. Electronic mar-

kets and electronic hierarchies. *Communications of the ACM*, 30(6):484–497, 1987. doi:10.1145/214762.214766.

[47] Jens Clausen, Birgit Blättel-Mink, Lorenz Erdmann, and Christine Henseling. Contribution of Online Trading of Used Goods to Resource Efficiency: An Empirical Study of eBay Users. *Sustainability*, 2(6):1810–1830, 2010. doi:10.3390/su2061810.