

Towards Fair Allocation in Social Commerce Platforms

Anjali Gupta

Indian Institute of Technology Delhi
India

Shreyans J. Nagori

Indian Institute of Technology Delhi
India

Abhijnan Chakraborty

Indian Institute of Technology Delhi
India

Rohit Vaish

Indian Institute of Technology Delhi
India

Sayan Ranu

Indian Institute of Technology Delhi
India

Prajit Prashant Nadkarni

Flipkart Internet Pvt Ltd
India

Narendra Varma Dasararaju

Flipkart Internet Pvt Ltd
India

Muthusamy Chelliah

Flipkart Internet Pvt Ltd
India

ABSTRACT

Social commerce platforms are emerging businesses where producers sell products through *re-sellers* who advertise the products to other customers in their social network. Due to the increasing popularity of this business model, thousands of small producers and re-sellers are starting to depend on these platforms for their livelihood; thus, it is important to provide *fair earning opportunities* to them. The enormous product space in such platforms prohibits manual search, and motivates the need for recommendation algorithms to effectively allocate product exposure and, consequently, earning opportunities. In this work, we focus on the fairness of such allocations in social commerce platforms and formulate the problem of assigning products to re-sellers as a fair division problem with indivisible items under *two-sided cardinality constraints*, wherein each product must be given to at least a certain number of re-sellers and each re-seller must get a certain number of products.

Our work systematically explores various well-studied benchmarks of fairness—including Nash social welfare, envy-freeness up to one item (*EF1*), and equitability up to one item (*EQ1*)—from both theoretical and experimental perspectives. We find that the existential and computational guarantees of these concepts known from the unconstrained setting do not extend to our constrained model. To address this limitation, we develop a mixed-integer linear program and other scalable heuristics that provide near-optimal approximation of Nash social welfare in simulated and real social commerce datasets. Overall, our work takes the first step towards achieving provable fairness alongside reasonable revenue guarantees on social commerce platforms.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Electronic commerce**; • **Theory of computation** → **Algorithmic mechanism design**.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '23, May 1–5, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9416-1/23/04...\$15.00
<https://doi.org/10.1145/3543507.3583398>

KEYWORDS

Social Commerce, Fair Division, Two-sided Constrained Allocation, Nash social welfare, Envy-Freeness, *EF1*, Equitability, *EQ1*

1 INTRODUCTION

Social commerce platforms are online platforms where social networks between users enable commerce. These platforms involve two primary stakeholders: *producers*, who sell their products on the platforms, and *re-sellers*, who are the users who help the producers reach out to the customers. In contrast to traditional e-commerce platforms where customers *directly* interact with the platform to purchase products, in social commerce the interaction is *indirect* and is facilitated by re-sellers who curate and promote products to other users/customers (see Figure 1). Re-sellers leverage their social networks (e.g., WhatsApp, Facebook, Instagram, as well as offline connections) to amplify the potential for sale [35]. Typically, the explicit social network being leveraged by the reseller is invisible to the social commerce platform.

Several companies around the world have successfully adopted social commerce as their business model. Prominent examples include Pinduoduo and Taobao in China, Meesho and Shopsy in India, Cafepress and Lockerz in USA, and Shopify in Canada. According to a recent study, social commerce industry is projected to reach USD 7.03 trillion globally by 2030 [46]. The impressive growth of social commerce stems from its ability to tap into a customer base that e-commerce has been unable to reach. Specifically, a large number of customers are still more comfortable with physical shopping than electronic marketplaces. This preference is due to various factors including a lack of digital skills, online payment issues, lack of trust over after-sales services, etc. [3]. Social commerce combines the benefits of e-commerce and physical stores: interacting with the re-sellers provides the customers with the same trust factor (the “human touch”) and comfort of interacting with physical stores (say, due to the ease of communication in the local dialect [6]) while extending the benefits of lower price and a larger item inventory commonly associated with electronic marketplaces.

By incorporating re-sellers in the business chain, social commerce provides livelihood to millions of people. For example, in India alone, one of the social commerce platforms engages 13 million re-sellers [45]. In addition, social commerce, much like e-commerce platforms, provides revenue opportunities to a large number of sellers/producers who sell products on these platforms. Since the number of products in the inventory is often huge, social commerce

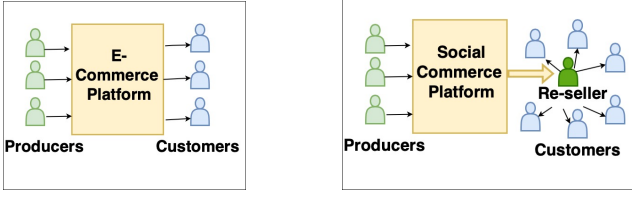


Figure 1: Producers connect with the customers directly in e-commerce platforms (left), whereas in social commerce, re-sellers facilitate this connection (right).

platforms assist re-sellers by recommending products that are more likely to be purchased during their social sharing [19, 54]. The recommendation engine, therefore, plays a crucial role in determining the revenues earned by both producers and re-sellers. Consequently, it is imperative that the recommendations are made in a *fair and equitable* manner to ensure a holistic long-term engagement of both parties with the platform.

While the fairness of recommendation in e-commerce platforms has been studied recently [1, 38, 42, 50], such questions remain unexplored in the context of social commerce. Furthermore, owing to the nature of the business model and resulting constraints in social commerce platforms, fairness notions from e-commerce do not trivially extend to social commerce. This gives rise to various challenges, as listed below:

- **Lack of customer visibility:** Many social commerce platforms do not have access to the social graphs among customers and re-sellers. In such cases, they rely entirely on the re-sellers' ability to promote and sell a product. This scenario is in sharp contrast to e-commerce platforms where the customers are directly visible and the recommendations can therefore be personalized. Thus, the social commerce recommendation, in effect, (virtually) allocates products to the re-sellers for selling. Our work makes the first attempt towards exploring fairness in such setting.
- **Re-seller constraints:** Each re-seller may have *expertise* only on a subset of products available in the inventory. Additionally, the number of allocations to each re-seller needs to be capped to ensure the design constraints of the UI, which allows showing only a certain number of products to each re-seller.
- **Product constraints:** Similarly, there are constraints on the product side as well. The platforms may have tie-ups with different producers demanding that every product get some minimum exposure – meaning that every product must be recommended to at least a few re-sellers.

Owing to the challenges outlined above, in this work, we seek to develop a *fair product allocation* algorithm for re-sellers which at the same time maximizes *cumulative revenue* of the re-sellers (as well as the social commerce platform) while providing some *minimum exposure guarantee* to each product on the platform. Our main contributions are summarized as follows:

- **Novel problem:** We formulate the problem of assigning products to re-sellers in social commerce platforms as a fair division problem with indivisible items under *two-sided cardinality constraints*. The product-side cardinality constraint ensures that each product must be allocated to at least a certain number of re-sellers. The

re-seller cardinality constraint, on the other hand, enforces that each re-seller must get a certain number of products (§ 2).

- **Theoretical analysis:** We explore various well-studied benchmarks of fairness—including *Nash social welfare*, *Envy-Freeness up to one item (EF1)*, and *Equitability up to one item (EQ1)*—in the context of two-sided cardinality constraints. We show that the positive existential and computational guarantees of these concepts in the unconstrained setting (i.e., without cardinality constraints) do not extend to our constrained setting (§ 3).
- **Algorithm design:** We develop a mixed-integer linear program (MILP) for the NP-hard problem of optimizing Nash social welfare in our setting. While the MILP is reasonable for small-sized simulated data, it turns out to be prohibitively slow for real-world datasets. We overcome this bottleneck by designing considerably faster heuristics that provide near-optimal efficacy (§ 4).
- **Empirical evaluation:** We perform extensive experiments on a large real-world social commerce dataset from Shopsy (owned by Flipkart), one of the largest social commerce platforms in India, to demonstrate the effectiveness and scalability of the proposed methodologies (§ 6).

2 PRODUCT ALLOCATION IN SOCIAL COMMERCE

In this section, we formulate the problem of product allocation in social commerce. First, we define the problem from a purely revenue maximization perspective. Then, we introduce the constraints from both producer and re-seller sides. In the next section, we modify the optimization objective to ensure fair revenue distribution.

2.1 Revenue Maximization Perspective

We begin with the assumption that there exists m re-sellers ($U = \{u_1, u_2, \dots, u_m\}$) and n products ($P = \{p_1, p_2, \dots, p_n\}$) in the social commerce platform.

Definition 1 (Expertise Matrix (E)). We assume that each re-seller has expertise in a specific set of products which we model via an expertise matrix $E \in \mathbb{R}^{m \times n}$ where $E_{i,j} \in [0, 1]$ represents the probability with which re-seller u_i sells a product p_j (after p_j is allocated to u_i).

In this work, we treat the construction of the expertise matrix as a black box. The typical approach to construct this matrix is through collaborative filtering [27, 29] or neural models [8, 25, 28].

Definition 2 (Utility (W)). We denote the expected revenue generated by re-seller u_i for product p_j as Utility $W_{i,j}$.

$$W_{i,j} = E_{i,j} \cdot \text{rev}(j) \quad (1)$$

Here $\text{rev}(j)$ is the revenue (or profit) generated from selling p_j .

Given a binary re-seller-product allocation matrix $A \in [0/1]^{n \times m}$, the expected revenue to the social platform, under the assumption of $W_{i,j}$ independence for all u_i and p_j , is:

$$E_{\text{rev}}(A) = \sum_{u_i \in U} \sum_{p_j \in P} W_{i,j} \times A_{i,j} \quad (2)$$

Hereon, we use the notation $A_i = \{p_j \in P, A_{i,j} = 1\}$ to denote the set of products allocated to re-seller u_i . Furthermore, we use $p_j \in A_i$ to denote $A_{i,j} = 1$.

A purely revenue maximization objective would therefore reduce to finding the allocation matrix A_{rev}^* that maximizes E_{rev} .

$$\arg \max_{A_{rev}^* \in \mathbb{A}} E_{rev}(A_{rev}^*) \quad (3)$$

Here, \mathbb{A} denotes the universe of all possible allocation matrices.

2.2 Two-Sided Constraints

While maximizing revenue is a reasonable goal, it does not consider some practical business requirements. For example, maximizing revenue may only allocate the most profitable products to the most successful re-sellers. As a result, other products may never get allocated, violating agreements a platform may have with the producers. Similarly, these platforms work within the design constraints of a UI, which typically shows a certain number of products to each re-seller. Without it, some re-sellers may get overwhelmed with product recommendations, while others may fail to get a reasonable number of products, hampering the re-seller experience. We introduce the following constraints to avoid these scenarios.

- (1) **RE-SELLER-CONSTRAINT**: Each re-seller gets allocated at least L_1 and at most L_2 unique products.

$$L_1 \leq |A_i| \leq L_2, \forall u_i \in U \quad (4)$$

- (2) **PRODUCT-CONSTRAINT**: Each product is allocated to at least R_1 and at most R_2 number of re-sellers.

$$R_1 \leq \sum_{i \in \{1, 2, \dots, m\}} \mathbb{1}\{p_j \in A_i\} \leq R_2, \forall p_j \in P \quad (5)$$

Next, we explore how we can achieve fair revenue distribution under these two constraints.

3 FAIR ALLOCATION UNDER CARDINALITY CONSTRAINTS

Prior research in other contexts has shown that focusing only on maximizing revenue can lead to unfairness [12, 34, 38, 42, 47]. Similarly, in social commerce, an algorithm maximizing revenue under the two-sided constraints may allocate all high revenue products to a few re-sellers, while others get only the low revenue ones. If most of the re-sellers do not find enough opportunities to earn well, they may leave a particular platform and go to its competitors. Thus, ensuring that resellers earn equitably can be a differentiating factor to attract more resellers into a platform and retain them in the long run. In this work, we aim to provide a *fair product allocation* scheme to re-sellers such that each re-seller is treated fairly in terms of earning opportunity and revenue generation.

To determine a suitable notion of fairness, we resort to the fair division literature [2] and consider two well-studied notions—envy-freeness and equitability—which are based on “intrapersonal” and “interpersonal” comparisons, respectively. For indivisible resources (as in our context), exact versions of these notions can not be guaranteed. Hence, one has to define approximations, and two popular approximations are envy-freeness up to one item (EF1), and equitability up to one item (EQ1). Next, we define EQ1 and EF1, and prove their non-existence in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5). Then, we discuss another well-studied measure of fairness—Nash social welfare—and establish its suitability in our context where we have to satisfy the two-sided cardinality constraints.

3.1 Equitability up to One Item (EQ1)

Definition 3 (Equitability up to one item (EQ1)). *An allocation $A = (A_1, \dots, A_n)$ is said to satisfy equitability up to one item (EQ1) if for every pair of re-sellers $u_i, u_k \in U$ such that $A_k \neq \emptyset$, there exists some item $p_j \in A_k$ such that $U_i(A_i) \geq U_k(A_k \setminus p_j)$.*

Intuitively, an EQ1 allocation ensures that the maximum revenue disparity between any pair of re-sellers is bounded by the revenue of one allotted item. For EQ1, positive existence results are known in the literature in the unconstrained setting.

Proposition 1. *EQ1 allocation always exists if an allocation is performed without any cardinality constraints [18, 23].*

However, in the presence of the two-sided constraints, an EQ1 allocation may not exist.

THEOREM 1. *EQ1 allocation may not exist in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).*

PROOF. Provided in Appendix A. \square

THEOREM 2. *Determining the existence of an EQ1 allocation in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5) is NP-complete.*

PROOF. Provided in Appendix B. \square

3.2 Envy-Freeness Up to One item (EF1)

Definition 4 (Envy-freeness up to one item (EF1)). *An allocation $A = (A_1, \dots, A_n)$ is said to satisfy envy-freeness up to one item (EF1) if for every pair of re-sellers $u_i, u_k \in U$ such that $A_k \neq \emptyset$, there is some item $p_j \in A_k$ such that $U_i(A_i) \geq U_i(A_k \setminus p_j)$. Here, $U_i(A_i) = \sum_{p_j \in A_i} W_{i,j}$, where $W_{i,j}$ is the utility of re-seller u_i for the item p_j .*

EF1 ensures that no re-seller envies another re-seller for more than one item. Specifically, if the items allotted to re-seller u_k (A_k) are instead allotted to u_i , then the maximum increase in the income of u_i is bounded by the revenue of one item in A_k . In the unconstrained setting, arbitrarily selecting any permutation of re-sellers and then allocating products greedily in round robin fashion gives EF1 solution [11]. We next show that in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5), the exact permutation of re-sellers matters.

Proposition 2. *Some permutation of re-sellers in round robin allocation may not find feasible EF1 solution in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).*

PROOF. Provided in Appendix A. \square

There are pseudopolynomial-time algorithms known for finding an EF1 and Pareto optimal¹ allocation without cardinality constraints [5] or cardinality constraints at product side [13]. However, their adaptation to our two-sided cardinality-constrained version is not straightforward. Freeman et al. [16] studied EF1 from both sides of matching parties (double envy-freeness up to one match) and concluded that two-sided EF1 i.e., double EF1, does not always

¹An allocation is Pareto optimal if no other allocation can make some re-seller strictly better off without making some other re-seller strictly worse off.

exist. Patro et al. [42] used round robin allocations to find *EF1* solution, but without strictly ensuring that product side constraints are satisfied. Gollapudi et al. [22] considered two-sided *EF1* in a dynamic setting, but only for symmetric binary valuations. In our case, however, the expected revenue can not be mapped to binary valuations.

3.3 Nash social welfare

In the unconstrained setting, the problem of maximizing the Nash social welfare is defined as follows.

Problem 1 (NASHOPTIMIZATION). *Given a set of n products $P = \{p_1, \dots, p_n\}$ and m re-sellers $U = \{u_1, \dots, u_m\}$ with utilities $\{W_{i,j}\}_{i \in [n], j \in [m]}$, the goal in NASHOPTIMIZATION is to return an allocation $A^* = (A_1^*, \dots, A_m^*)$ that maximizes the geometric mean (equivalently, the product) of agents' utilities. That is,*

$$A^* \in \operatorname{argmax}_{A=(A_1, \dots, A_m)} \prod_{i \in [m]} \sum_{p_j \in A_i} W_{i,j} \quad (6)$$

where $A_i \cap A_k = \emptyset \forall i, k \in [m]$ and $A_1 \cup \dots \cup A_m = P$.

Replacing revenue with Nash social welfare in the objective function hits a sweet spot between Bentham's utilitarian notion of social welfare (maximize the sum of utilities) and the egalitarian notion of Rawls (maximize the minimum utility) in unconstrained cardinality setting[11].

While *EQ1* and *EF1* are constraint satisfaction based objectives specifying whether an allocation is permitted or not, NASHOPTIMIZATION is an optimization problem. Historically, NASHOPTIMIZATION has mostly been studied in a setting where only one copy of each product is available for allocation. Chaudhury et al. [13] proposed a polynomial-time algorithm to approximate the optimal Nash social welfare (NASHOPTIMIZATION) up to a factor of $e^{1/e}$ considering the product side cardinality; however, there is no other constraints (on the re-seller side cardinality). We focus on optimizing NASHOPTIMIZATION in two-sided cardinality setting in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).

Problem 2. Two-sided cardinality constrained Nash Optimization (2S-CARDNASHOPT): *Given the set of m re-sellers ($U = \{u_1, \dots, u_m\}$), n products ($P = \{p_1, \dots, p_n\}$), and the parameters L_1, L_2, R_1, R_2 of the RE-SELLER-CONSTRAINT and PRODUCT-CONSTRAINT, identify the allocation matrix \mathcal{A} maximizing:*

$$\mathcal{A} = \operatorname{argmax}_A \prod_{u_i \in U} \sum_{p_j \in A_i} W_{i,j} \quad (7)$$

subject to satisfying RE-SELLER-CONSTRAINT and PRODUCT-CONSTRAINT.

This introduces non-trivial challenges and highlights why obtaining fairness in social commerce is a challenging problem. To rigorously establish this, we next discuss the known results for NASHOPTIMIZATION in the literature, their validity under two-sided constraints, and why NASHOPTIMIZATION is the objective function of choice for the proposed problem.

It is known that allocations maximizing Nash social welfare simultaneously satisfy *EF1* and Pareto optimality—a well-studied criterion of economic efficiency in an unconstrained setting [11]. However, the same result may not hold in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).

THEOREM 3. *Optimising Nash social welfare (Eq. 7) may not give *EF1* solution in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).*

PROOF. Provided in Appendix A. \square

Next, we prove that two-sided cardinality constrained NASHOPTIMIZATION is NP-hard, which makes the proposed problem NP-hard.

THEOREM 4. *Two-sided cardinality constrained Nash Optimization (2S-CARDNASHOPT) problem is NP-hard.*

PROOF. Provided in Appendix B. \square

3.4 Suitability of Optimizing Nash social welfare for Social Commerce

EQ1 allocation under two-sided cardinality constraints may not exist (Theorem 1). Furthermore, determining the existence itself is NP-complete (Theorem. 2). Similarly, double-sided *EF1* may not exist either [16]. The feasibility of *EF1* under two-sided cardinality constraints depends upon the selection of re-seller permutation in round-robin allocation (proposition 2). In contrast, although 2S-CARDNASHOPT is NP-hard, we can attempt to design heuristics and identify allocations with high Nash social welfare. This motivates us to choose Nash social welfare as the fairness measure of choice.

Note that by maximizing the (mathematical) product of the individual revenues obtained by each re-seller, we value allocation schemes higher where none of the individual re-seller revenues is low. This ensures re-seller-side fairness. Fairness on the exposure of products is obtained through PRODUCT-CONSTRAINT. As we show later, the proposed heuristic for 2S-CARDNASHOPT provides near-optimal performance empirically. In the next section, we design MILP and our heuristics for 2S-CARDNASHOPT.

4 PROPOSED ALGORITHMS

Next, we describe the proposed algorithms to optimize Nash social welfare under two-sided cardinality constraints. Our contributions towards this goal include a mixed integer linear program (§ 4.1) which is an adaptation of an existing MILP from the unconstrained setting [11], and two iterative greedy heuristics (§ 4.2).

4.1 Mixed Integer Linear Programming (MILP)

Since maximizing Nash social welfare in the unconstrained fair division problem is APX-hard [32], this motivates the development of integer linear programs and heuristics to work on small instances. To this end, Caragiannis et al. [11] proposed a mixed integer linear program (MILP) that was based on the idea of lower bounding the log function by a piecewise linear function.² The latter was designed to be exactly equal to the log function on integral points, and thus a piecewise linear approximation suffices. We adapt this MILP to our setup by incorporating two-sided cardinality constraints and call the adapted program NASHMAX.

²Notice that maximizing the Nash welfare product is equivalent to maximizing the sum of log of the utilities.

4.1.1 NASHMAX. We scale all $W_{i,j}$ (utility) between 1 to 1000 integers and let those scaled integer values are $v_{i,j}$, $\forall i$ and j such that $\sum_j v_{i,j} \leq 1000, \forall i$. Let γ_i be a continuous variable denoting the log of the utility to player i , and bound it using a set of linear constraints such that the tightest bound at every integral point k is exactly $\log(k)$.

$$\text{Maximize } \sum_{i \in U} \gamma_i \quad (8)$$

Subject to

- (1) Log approximation: The RHS in the constraint below is a lower bound on the log function everywhere and is tight on all points where the utility is integral.

$$\forall i, \gamma_i \leq \log k + [\log(k+1) - \log k] \times \left[\sum_{j \in P} x_{i,j} \times v_{i,j} - k \right],$$

$$\forall i \in U, k \in \{1, 3, \dots, 999\}$$

- (2) Checking whether item j is assigned to re-seller i

$$x_{i,j} \in \{0, 1\}$$

- (3) Each item is allocated to at least R_1 re-sellers and at most R_2

$$\forall j, R_1 \leq \sum_i x_{i,j} \leq R_2$$

- (4) Each re-seller is allocated at least L_1 and at most L_2 items

$$\forall i, L_1 \leq \sum_j x_{i,j} \leq L_2$$

Proposition 3. *The allocation returned by NASHMAX has the highest Nash social welfare among all cardinality-constrained allocations for the given instance.*

We note that the proof of correctness of Proposition 3 is similar to that of the unconstrained version as discussed in [11] and is therefore omitted. Unfortunately, the MILP described above does not scale to large datasets. This motivates us to develop greedy heuristics for maximizing Nash welfare.

4.2 Iterative Greedy Heuristics

Here we propose two heuristics based approaches to optimize Nash under two-sided cardinality constraints. The first approach is based on pure greedy allocation, where we iteratively allocate each product to a re-seller which maximizes the Nash. The second approach is round wise allocation, where the poorest re-seller is preferred to pick up its most preferred product in each round.

4.2.1 GREEDYNASH. We now describe an iterative greedy approach that we call GREEDYNASH (see Algorithm 1). The underlying idea is to allocate each product to myopically achieve the maximum increase in Nash social welfare. Specifically, starting from an empty allocation (lines 1-2), we first allocate each re-seller exactly one product by allowing the re-sellers to pick their most preferred product in a round-robin manner subject to maintaining cardinality constraints (lines 3-4). This step ensures that each re-seller acquires a non-zero utility (thus, in turn, making the Nash product non-zero) and makes the remaining decisions of the algorithm well-defined.

Next, we iterate over the set of products, and, at each step, assign each product in a way that maximizes the marginal increase in Nash Welfare subject to the cardinality constraints (lines 5-8).

Algorithm 1 GREEDYNASH: Greedily Maximize Nash under Two-Sided Cardinality Constraints

Input: m re-sellers, n products, utilities $W_{i,j}$, $i \leq m$, $j \leq n$, and parameters L_1 and L_2 (The lower and upper bound on recommendation list for re-sellers), R_1 and R_2 (Minimum and maximum number of re-sellers allocated to each product respectively).
Output: Assignments of re-sellers $A = \{A_1, A_2, \dots, A_m\}$.

```

1: for  $i = 1$  to  $m$  do:
2:   set  $A_i = \{\}$ , (Empty allocations for re-seller  $u_i$ )

--- FIRST ALLOCATION ---
3: for  $i = 1$  to  $m$  do:
4:   Assign re-seller  $u_i$ 's most preferred unique product,  $p_j$  to  $A_i$  subject to cardinality constraints

--- GREEDY ALLOCATION ---
5: for  $j = 1$  to  $n$  do:
6:   repeat
7:     Find re-seller  $u_i$  such that,  $p_j \notin A_i$ , and allocation of product  $p_j$  to re-seller  $u_i$  leads to maximum increase in Nash subject to cardinality constraints (with lower bounds). Assign product  $p_j$  to  $A_i$ 
8:   until product  $p_j$  is allocated to  $R_1$  users OR no such suitable user found

--- FULL-FILL RECOMMENDATION LIST ---
9: Some users may not be allocated to  $L_1$  products yet. Allocate their most preferred product until their allocation list contains  $L_1$  unique products subject to the most preferred product should not be already allocated to  $R_2$  users.

--- GREEDY REPLACEMENT ---
10: for each products  $p_j$  where number of re-sellers allocated to it is less than  $R_1$  do:
11:   repeat
12:     Find re-seller  $u_i$  such that,  $p_l \in A_i$  with number of re-sellers allocated to  $p_l > R_1$ ,  $p_j \notin A_i$ , and replacement of  $p_l$  with  $p_j$  result in a minimum decrease in her/his utility
13:     Replace  $p_l$  with  $p_j$  in  $A_i$ 
14:   until Product  $p_j$  is allocated to  $R_1$  users

--- ASSIGN GREEDILY TILL UPPER-BOUNDS SATISFIED ---
15: for  $j = 1$  to  $n$  do:
16:   repeat
17:     Find re-seller  $u_i$  such that,  $p_j \notin A_i$ , and allocation of product  $p_j$  to re-seller  $u_i$  leads to maximum increase in Nash subject to cardinality constraints (with upper bounds). Assign product  $p_j$  to  $A_i$ 
18:   until product  $p_j$  is allocated to  $R_2$  users OR no such suitable user found

```

This process continues until either all R_1 copies of a product have been allocated or no suitable user is found (line 8). If the cardinality constraint for one or more users is not met, we continue allocating them their most-preferred products (line 9). Similarly, products not achieving their minimum cardinality are greedily assigned to re-sellers (lines 10-14). Once all products achieve their minimum cardinality, we can greedily allocate R_2 copies of products subject to user-side cardinality constraints (lines 15-18).

4.2.2 SEAL. It is a sequential egalitarian algorithm (Algorithm 2). Initially, allocations are empty; hence the utility of all re-sellers is 0. We perform allocations in L_1 round. Each round sequentially allocates the poorest re-seller (with the least utility) their most preferred product subject to lower bound cardinality constraints. At the end of L_1 rounds, if some products are not allocated to R_1 re-sellers, perform greedy product replacement as mentioned in steps 12-16. Once every product achieves their minimum cardinality, we can allocate R_2 copies of products subject to user-side cardinality constraints (steps 17-22).

Between GREEDYNASH and SEAL, the outperformance depends on the utility values of the re-sellers (Appendix D provides some relevant Examples). Computational complexity of both algorithms are discussed in Appendix C. Note that in both cases, a feasible solution may not exist under all possible cardinality constraints. However, the constraints can be set to ranges that guarantee feasibility (§ 6).

Algorithm 2 SEAL: Sequential Egalitarian Algorithm under Two-Sided Cardinality Constraints

Input: m re-sellers, n products, utilities W_{ij} , $i \leq m$, $j \leq n$, and parameters L_1 and L_2 (The lower and upper bound on recommendation list for re-sellers), R_1 and R_2 (Minimum and maximum number of re-sellers allocated to each product, respectively).

Output: Assignments of re-sellers $A = \{A_1, A_2, \dots, A_m\}$.

```

----- INITIALIZATION -----
1:  $PQ = []$  (Empty ascending order priority queue)
2: for  $i = 1$  to  $m$  do:
3:    $set A_i = \{\}$ , (Empty allocations for re-seller  $u_i$ )
4:   Insert pair  $\langle u_i, 0 \rangle$  i.e.  $\langle \text{user } u_i, u_i \text{'s utility} \rangle$  to  $PQ$ 

----- SEQUENTIAL ITERATIVE GREEDY ALLOCATION WITH LOWER BOUND -----
5: for  $k = 1$  to  $L_1$  do:
6:   while  $PQ$  is empty do
7:     Take out a reseller  $u_i$  with minimum utility from  $PQ$ 
8:     Assign re-seller  $u_i$  to his/her most preferred product,  $p_j$  such that  $p_j \notin A_i$  and the
       number of re-sellers allocated to  $p_j$  is less than  $R_1$ 
9:     If no such  $p_j$  is found then assign re-seller  $u_i$  to his/her most preferred product,  $p_j$ 
       such that  $p_j \notin A_i$  and the number of re-sellers allocated to  $p_j$  is less than  $R_2$ 
10:  for  $i = 1$  to  $m$  do:
11:    Insert pair  $\langle u_i, u_i \text{'s utility} \rangle$  to  $PQ$ 

----- GREEDY REPLACEMENT -----
12: for each products  $p_j$  where number of re-sellers allocated to it is less than  $R_1$  do:
13:   repeat
14:     Find re-seller  $u_i$  such that,  $p_l \in A_i$  with number of re-sellers allocated to  $p_l > R_1$ ,
        $p_j \notin A_i$ , and replacement of  $p_l$  with  $p_j$  result in a minimum decrease in her/his utility
15:     Replace  $p_l$  with  $p_j$  in  $A_i$ 
16:   until Product  $p_j$  is allocated to  $R_1$  users

----- SEQUENTIAL ITERATIVE GREEDY ALLOCATION WITH UPPER BOUND -----
17: for  $k = L_1$  to  $L_2$  do:
18:   while  $PQ$  is empty do
19:     Take out a reseller  $u_i$  with minimum utility from  $PQ$ 
20:     Assign re-seller  $u_i$  to his/her most preferred product,  $p_j$  such that  $p_j \notin A_i$  and the
       number of re-sellers allocated to  $p_j$  is less than  $R_2$ 
21:   for  $i = 1$  to  $m$  do:
22:     Insert pair  $\langle u_i, u_i \text{'s utility} \rangle$  to  $PQ$ 

```

5 DATASETS

In this section, we introduce the datasets used to benchmark the proposed methodologies.

5.1 Real Social Commerce Dataset

The dataset from Shopsy comprises of one month of click and purchases history, followed by the next six months' purchase history. In addition, it includes the product catalog, which details the price, revenue (profit), and category of each product. Table 1 presents the statistics of this dataset. It is worth noting that this is possibly the largest real dataset being studied in the fair allocation literature.

Since the dataset contains products of various price-ranges and categories, it is not fair to compare the revenues generated by re-sellers across different categories with vastly different expertise. As an example, re-sellers specializing in expensive electronic items (Ex: TV, laptop, etc.) have significantly different expectations in terms of product allocations and revenue contribution to platform when compared to those who specialize in garments. Hence, we partition re-sellers into homogeneous clusters by performing k -means clustering on their expertise vectors. The expertise vectors are learned using collaborative filtering [29]. The value of k is set to 53 since inter-cluster distance computed through elbow-method [39] stabilizes at this value. The problem of 2S-CARDNASHOPT is solved on each cluster separately, and the results are reported in § 6. The distributions of the cluster sizes and the average price of items sold by re-sellers in each cluster are shown in Fig. 2. As can be seen, there is a significant disparity in the price of the items sold by each cluster, which necessitates this segregation step of the raw data through clustering.

#Re-sellers	#Products	#Purchases	#Product categories
531933	13178	951664	703

Table 1: Summary of the real social commerce dataset.

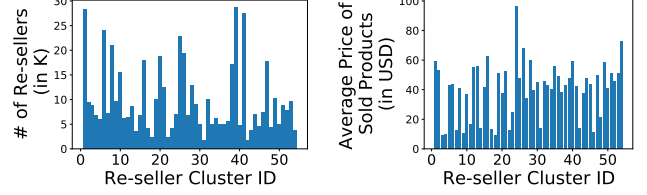


Figure 2: Distributions of the re-sellers (left) and the average price of products sold by them in each cluster (right).

5.2 Synthetic Dataset

Since MILP (Recall § 4.1) does not scale to large datasets, we create small, synthetic datasets so that the proposed heuristic can be compared to the optimal allocation. More details about the synthetic dataset generation is provided in Appendix F.

6 EXPERIMENTAL RESULTS

In this section, we benchmark the proposed methodologies on the datasets described in § 5 and establish that:

- **Approximation quality:** SEAL achieves near optimal performance with minimal dip in revenue.
- **Efficacy:** SEAL provides significantly higher fairness while also ensuring minimal reduction in the revenue when compared to purely maximizing revenue.

The implementations of all algorithms used in this study are in Python 3.0. Our experiments are performed on a machine with Intel(R) Xeon(R) CPU @ 2.10GHz with 252GB RAM on Ubuntu 18.04.3 LTS. Our codebase is available at <https://github.com/idea-iitd/FairAllocInSocialCom.git>

6.1 Baselines

To contextualize the performance of the proposed algorithms, we compare its performance to the following baselines:

- **REVMAX:** We use Integer Linear Programming (ILP) to maximize the overall revenue (Eq. 2) by enforcing both RE-SELLER-CONSTRAINT and PRODUCT-CONSTRAINT.
- **GREEDYREVENUE:** We use the iterative greedy approach to maximize revenue in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5). Specifically, it proceeds in an iterative manner and in each iteration we select the $\langle \text{re-seller}, \text{product} \rangle$ tuple that maximizes revenue without violating either RE-SELLER-CONSTRAINT or PRODUCT-CONSTRAINT. While REVMAX is guaranteed to provide a superior revenue, GREEDYREVENUE is significantly faster and hence scales to large datasets.
- **FAIRREC** [42] and **TFROM** [52] are two-sided fairness-aware recommendation algorithms. FAIRREC ensures EF1 allocations for customers, and aims to achieve product side constraints. TFROM considers the position bias in recommendations.
- **FOEIR** [44] is fair-ranking algorithm that deal with advantaged and disadvantaged social groups' bias and positional bias. In our adaptation, we randomly divided products into these two groups,

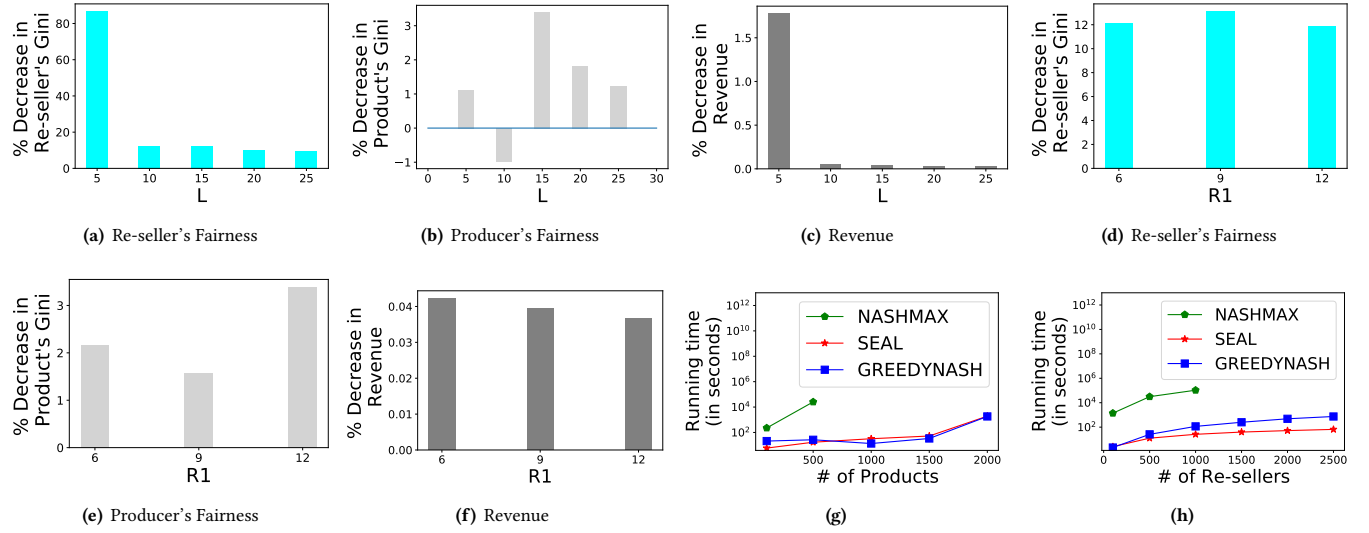


Figure 3: The bar plot shows (a) an increase in fairness among re-sellers, (b) a decrease/increase in fairness among products, and (c) a decrease in revenue on comparing NASHFAIR to REVMAX w.r.t. variation in L . Recall that L_1 and L_2 are set as $L - \epsilon$ and $L + \epsilon$ respectively where $\epsilon = 3$. (d-f) represent the same metrics but, w.r.t. $R1$. (g-h) Scalability: Running time (in seconds) comparison of SEAL, GREEDYNASH, and NASHMAX on synthetic dataset w.r.t. increase in the number of (g) products and (h) re-sellers.

assumed a uniform attention model and applied the FOEIR algorithm maintaining the cardinality constraints.

In addition to the baselines, this work contributes three algorithms, namely NASHMAX, which optimizes Nash through MILP (§ 4.1), GREEDYNASH (Alg. 1) and SEAL (Alg. 2). Also note that we do not consider revenue maximization without constraints, since such allocation results in the highest priced product being allocated to all re-sellers, which is impractical.

6.2 Parameters

Real dataset: We set $L_1 = L - \epsilon$ and $L_2 = L + \epsilon$ for $\epsilon = 0$ and $L = 15$ in RE-SELLER-CONSTRAINT, and $R_1 = \lfloor \frac{L_1 + L_2}{2} \times \frac{m}{n} \rfloor$, $R_2 = m$ in PRODUCT-CONSTRAINT. Here, n and m represent the number of products and re-sellers respectively. **Synthetic dataset:** We vary $L = [5, 10, 15, 20, 25]$, where $L_1 = L - \epsilon$ and $L_2 = L + \epsilon$ for $\epsilon = 3$. We set the minimum guaranteed copy for each product to R_1 , i.e., $R_1 = \lfloor \alpha \times L_1 \times m/n \rfloor$ for $\alpha = [0.5, 0.75, 1]$ and $R_2 = \lfloor m, 2 \times R_1 \rfloor$.

6.3 Metrics

We quantify performance using the following metrics.

- **Revenue:** The total expected revenue from all products (Eq. 2).
- **Gini coefficient:** The Gini coefficient is the ratio of the area that lies between the line of equality and the Lorenz curve over the total area under the line of equality [20]. Mathematically,

$$Gini = \frac{\sum_{i=1}^m \sum_{j=1}^m |U_i - U_j|}{2m \sum_{j=1}^m U_j} \quad (9)$$

where U_i is the expected revenue (utility) of re-seller i and n is the number of re-sellers for computing fairness for re-sellers. For products, i is product and n is number of products. A lower Gini indicates fairer distribution.

- **Average income gap:** The income gap is the difference between the maximum expected revenue with the minimum one across all re-sellers, i.e.,

$$\text{Income Gap} = \max_{\forall u_i, u_k \in U} \{|rev(u_i) - rev(u_k)|\}, \quad (10)$$

where $rev(u_i) = \sum_{p_j \in A_i} W_{i,j}$.

- **Percentage of constraint-satisfying allocations:** If we are given X datasets/clusters of re-sellers and the corresponding products to be allocated, we find in what percentage of the datasets/clusters, an algorithm is able to provide an allocation satisfying RE-SELLER-CONSTRAINT and PRODUCT-CONSTRAINT.

6.4 Comparison of the Optimal Approaches

The MILP formulations, i.e., REVMAX, CONREVMAX and FAIRNASH provide optimal allocations under integer valuations. However, since MILP-based algorithms are prohibitively slow, we measure their efficacy only on the synthetic datasets.

6.4.1 Efficacy. Fig. 3 presents the results. We observe that enforcing Nash has minimal impact on the revenue. Furthermore the fairness on re-seller side in terms of Gini is significantly better. The Gini on the product sides are similar for REVMAX and NASHMAX since both ensure PRODUCT-CONSTRAINT.

6.4.2 Scalability. As noted earlier MILP-based algorithms are prohibitively slow, making them impractical for real datasets. We substantiate this claim by analyzing the growth of their running times against the number of re-sellers and products in the system. Figs. 3(g)-3(h) present the results, where we compare NASHMAX with the greedy heuristics of GREEDYNASH and SEAL. As visible, the greedy heuristics are more than 2 orders of magnitude faster.

	SEAL	GREEDYNASH
Avg. approx. ratio	0.9989	0.997
Min. approx. ratio	0.9952	0.987
% Dip in revenue	0.47	1.61
Avg. income gap ratio	0.97	1.02

Table 2: Comparison of SEAL and GREEDYNASH on synthetic datasets. We present the ratios when compared to the same metric obtained by NASHMAX.

6.5 Efficacy of Greedy Heuristics

The fast running times of greedy heuristics make them an attractive proposition. We therefore investigate their efficacy on the synthetic dataset against NASHMAX, which is optimal in terms of Nash.

6.5.1 Performance on synthetic datasets. Table 2 presents the results. We observe that the approximation ratios of GREEDYNASH and SEAL are ≈ 0.99 indicating excellent efficacy; SEAL performs marginally better. We also report the minimum approximation ratios across all instances of synthetic datasets and notice that it is close to the average indicating excellent stability. The dip in revenue is less than 2%, with SEAL being the better performer. Finally, the average of the income gaps is also close to the optimal. Overall, these results establish greedy heuristics as a good alternative due to obtaining excellent balance between efficiency and efficacy.

6.5.2 Performance on real datasets: We next measure performance on the real datasets. Although no technique consistently achieves the best performance across all metrics, SEAL and GREEDYNASH achieve the best overall balance in performance. We notice that both FAIRREC and TFROM violate cardinality constraints on the product side in at least 96% of the instances. Although these baseline algorithms try to adhere to two-sided constraints, they do not guarantee adherence, which we do. This property manifests itself in the trends visible in Table 3. Finally, although TFROM achieves the best Gini on the re-seller side, its revenue is significantly worse, making it impractical for social commerce platforms.

7 RELATED WORK

Social recommendation: In a social network, users spread their preferences for items to their connections by sharing the items, and this potentially converts to the sales of items among connections. In the social commerce domain, the problem of Influence Maximization (IM) is well-studied [15, 24, 30, 31, 33, 41, 48, 49]. The focal point of IM is, however, different from ours. In IM, given a social network and a budget b , the goal is to select n nodes from the network, such that if they advertise your product, then the cascading effect of this action is maximized through network diffusion. In our problem, the set of influencers i.e., re-sellers, are given to us as input. Thus, our focus is on which influencer i.e., re-sellers, should promote which product, which is the allocation aspect. In addition, the social network between re-sellers and customers is unavailable in our case. Hence, recommendation techniques based on the rating of goods provided by users and the interaction of those users in a social network [14, 19, 35–37, 54, 55] is out of our scope.

Allocation problems: Fair allocation has been widely studied in computational social choice theory, where the primary interest is in the allocation of divisible goods (aka the cake-cutting problem).

Metric	SEAL	GREEDYNASH	FAIRREC	GREEDY-REVENUE	TFROM	FOEIR
Revenue (in USD)	381k	372k	349k	390k	167k	338.9k
Avg. gini (users)	0.314	0.273	0.43	0.553	0.23	0.43
Avg. income gap	672.11	408	1123.257	1525	438	1338.01
% of Cardinality-violations	0	0	96.22	0	100	0

Table 3: Comparison of SEAL and GREEDYNASH with baselines on real social commerce dataset. The best result for each metric (row) is highlighted in bold.

Here, however, we are in the realm of indivisible goods since our allocation is binary – a product can either be allocated or not. Very recently, some papers focused on allocating indivisible goods in budgeted course allocation [10], balanced graph partition [9], or allocation of cardinality constrained group of resources [7]. Few other sets of papers propose envy freeness up to one good (EF1) and Pareto optimal product allocations [4, 17, 42, 53]. Going beyond, we explore various well-studied notions of fairness in the allocation of goods –including Nash social welfare, EF1, and equitability up to one item (EQ1)—in the context of two-sided cardinality constraints.

Fair Recommendation: Wu et al. [50] proposes a multi-objective gradient descent algorithm that optimizes fairness and utility to balance the objectives between consumers and producers. Few recent works propose a framework to learn the relevance scores considering the joint multi-objective optimization [21, 51]. However, in this work, we only focus on the allocation aspect. To the best of our knowledge, this is the first work on the allocation problem in the context of social commerce, which does not have a social graph among re-sellers and customers.

8 CONCLUSION

Social commerce has emerged as a successful model to doing business. Recommendation engines form a core component of the social commerce platform and dictates the revenue earned by both producers and re-sellers. While several works have studied fairness issues on e-commerce recommendation engines, it remains unexplored in the context of social commerce. In this work, we bridged this gap. Our analysis revealed that ensuring fairness on social commerce maps to a special case of fair division of individual goods under two-sided cardinality constraints. We studied multiple fairness notions and established that many of the existential and computational guarantees of unconstrained setting do not extend to two-sided cardinality settings. Based on our discoveries, we proposed Nash social welfare the optimization function of choice. We showed that optimizing Nash under two-sided constraints is NP-hard, and hence designed polynomial-time greedy heuristics to overcome the bottleneck. Extensive experiments on real datasets obtained from a leading social commerce firm showed that the proposed heuristics provide near-optimal efficacy in a fraction of time. In addition, they out-performed established baseline algorithms showcasing the need to design a specialized algorithm that is cognizant of the social commerce context.

Acknowledgements: A. Gupta gratefully acknowledges the support of Google Ph.D. Fellowship. R. Vaish acknowledges support from DST INSPIRE grant no. DST/INSPIRE/04/2020/000107 and SERB grant no. CRG/2022/002621.

REFERENCES

- [1] Himan Abdollahpour and Robin Burke. 2019. Multi-stakeholder Recommendation and its Connection to Multi-sided Fairness. In *RMSE 2019*.
- [2] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. 2022. Fair Division of Indivisible Goods: A Survey.
- [3] Nina Angelovska. 2018. 6 Reasons Why Europeans Don't Shop Online. <https://www.forbes.com/sites/ninaangelovska/2018/10/23/6-reasons-why-europeans-dont-shop-online/?sh=10fb6bb22869>.
- [4] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. 2019. Fair Allocation of Indivisible Goods and Chores. In *Proc. IJCAI*.
- [5] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. 2018. Finding Fair and Efficient Allocations. In *Proc. of EC*. 557–574.
- [6] Tarush Bhalla. 2020. Social commerce firms focus on local tongue. <https://www.livemint.com/technology/tech-news/social-commerce-firms-focus-on-local-tongue-11596811490216.html>.
- [7] Arpita Biswas and Siddharth Barman. 2018. Fair Division under Cardinality Constraints. In *Proc. IJCAI*.
- [8] Jesús Bobadilla, Raúl Lara-Cabrera, Ángel González-Prieto, and Fernando Ortega. 2020. DeepFair: Deep Learning for Improving Fairness in Recommender Systems. *CoRR* abs/2006.05255 (2020). arXiv:2006.05255
- [9] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. 2017. Fair Division of a Graph. arXiv:1705.10239 [cs.GT]
- [10] Eric Budish. 2010. The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. In *BQGT*.
- [11] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel Procaccia, and Junxing Wang. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation* 7 (09 2019), 1–32.
- [12] Abhijnan Chakraborty, Anikó Hannák, Asia J. Biega, and K. Gummadi. 2017. Fair Sharing for Sharing Economy Platforms. In *Proc. FATREC*.
- [13] Bhaskar Chaudhury, Yun Cheung, Jugal Garg, Naveen Garg, Martin Hoefer, and Kurt Mehlhorn. 2022. Fair Division of Indivisible Goods for a Class of Concave Valuations. *Journal of Artificial Intelligence Research* 74 (05 2022), 111–142.
- [14] Rui Chen, Yan-Shuo Chang, Qingyi Hua, Quanli Gao, Xiang Ji, and Bo Wang. 2020. An enhanced social matrix factorization model for recommendation based on social networks using social interaction factors. *Multimedia Tools and Applications* 79 (05 2020), 1–31.
- [15] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. *Proc. ICDM*, 88–97.
- [16] Rupert Freeman, Evi Micha, and Nisarg Shah. 2021. Two-Sided Matching Meets Fair Division. In *Proc. IJCAI*, Zhi-Hua Zhou (Ed.). 203–209.
- [17] Rupert Freeman, Nisarg Shah, and Rohit Vaish. 2020. Best of Both Worlds: Ex-Ante and Ex-Post Fairness in Resource Allocation.
- [18] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2019. Equitable Allocations of Indivisible Goods.
- [19] Chen Gao, Chao Huang, Donghan Yu, Hao hao Fu, Tzu-Heng Lin, Depeng Jin, and Yong Li. 2020. Item Recommendation for Word-of-Mouth Scenario in Social E-Commerce. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [20] Joseph L. Gastwirth. 1972. The estimation of the Lorenz curve and Gini index. *The review of economics and statistics* (1972), 306–316.
- [21] Yingqiang Ge, Xiaoting Zhao, Lucia Yu, Saurabh Paul, Diane Hu, Chu-Cheng Hsieh, and Yongfeng Zhang. 2022. Toward Pareto Efficient Fairness-Utility Trade-off in Recommendation through Reinforcement Learning. In *Proc. WSDM*. 316–324.
- [22] Sreenivas Gollapudi, Kostas Kollias, and Benjamin Plaut. 2020. Almost Envy-Free Repeated Matching In Two-Sided Markets. In *WINE* (Beijing, China). Springer-Verlag, Berlin, Heidelberg, 3–16.
- [23] Laurent Gourvès, Jérôme Monnot, and Lydia Tilane. 2014. Near Fairness in Matroids. In *Proc. ECAI*. 393–398.
- [24] Amit Goyal, Wei Lu, and Laks Lakshmanan. 2011. SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. *Proc. ICDM*, 211–220.
- [25] Qianxiu Hao, Qianqian Xu, Zhiyong Yang, and Qingming Huang. 2021. Pareto Optimality for Fairness-Constrained Collaborative Filtering. In *Proc. ACM International Conference on Multimedia*. 5619–5627.
- [26] Juris Hartmanis. 1982. Computers and Intractability: A Guide to the Theory of NP-Completeness (Michael R. Garey and David S. Johnson). *SIAM Rev.* 24, 1 (1982), 90–91.
- [27] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proc. ICDM*. 263–272.
- [28] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. 2021. Debiasing Career Recommendations with Neural Fair Collaborative Filtering. In *Proc. WWW*. 3779–3790.
- [29] Christopher C. Johnson. 2014. Logistic Matrix Factorization for Implicit Feedback Data.
- [30] Kyomin Jung, Wooram Heo, and Wei Chen. 2011. IRIE: Scalable and Robust Influence Maximization in Social Networks. *Proc. ICDM* (11 2011).
- [31] V. Kumar, Vikram Bhaskaran, Rohan Mirchandani, and Milap Shah. 2013. Practice Prize Winner—Creating a Measurable Social Media Marketing Strategy: Increasing the Value and ROI of Intangibles and Tangibles for Hokey Pokey. *Marketing Science* 32 (03 2013), 194–212.
- [32] Euiwoong Lee. 2017. APX-Hardness of Maximizing Nash Social Welfare with Indivisible Items. *Inform. Process. Lett.* 122 (2017), 17–20.
- [33] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne Vanbriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. *Proc. SIGKDD* 420–429, 420–429.
- [34] Nixie S Lesmana, Xuan Zhang, and Xiaohui Bei. 2019. Balancing Efficiency and Fairness in On-Demand Ridesourcing. In *Proc. NeurIPS*, Vol. 32.
- [35] Tzu-Heng Lin, Chen Gao, and Yong Li. 2019. Cross-platform recommendation for social e-commerce. In *Proc. SIGIR*. 515–524.
- [36] Xiaoling Long, Chao Huang, Yong Xu, Huance Xu, Peng Dai, Lianghao Xia, and Liefeng Bo. 2021. Social recommendation with self-supervised metagraph informax network. In *Proc. CIKM*. 1160–1169.
- [37] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Proc. CIKM*. 931–940.
- [38] Masoud Mansoury. 2021. Fairness-Aware Recommendation in Multi-Sided Platforms. In *Proc. WSDM*. 1117–1118.
- [39] Rena Nainggolan, Resiatna Perangin-angin, Emma Simarmata, and Astuti Feriani Tarigan. 2019. Improved the performance of the K-means cluster using the sum of squared error (SSE) optimized by using the Elbow method. In *Journal of Physics: Conference Series*, Vol. 1361.
- [40] Trung Thanh Nguyen et al. 2022. An Experimental Study of Fast Greedy Algorithms for Fair Allocation Problems. *Journal on Information Technologies & Communications* 2022, 2 (2022), 71–81.
- [41] N. Ohsaka, T. Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations. *Proceedings of the National Conference on Artificial Intelligence* 1 (01 2014), 138–144.
- [42] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms. In *Proc. WWW* (Taipei, Taiwan) (WWW '20). 1194–1204.
- [43] Erel Segal-Halevi. 2019. Fair division with bounded sharing. *arXiv preprint arXiv:1912.00459* (2019).
- [44] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. *CoRR* abs/1802.07281 (2018). arXiv:1802.07281
- [45] Manish Singh. 2021. Indian social commerce Meesho raises \$570 million at \$4.9 billion valuation. <https://techcrunch.com/2021/09/29/meesho-india-social-commerce-raises-570-million/>.
- [46] strategicmarketresearch. 2022. Social Commerce Market Size, Share, Global Report 2030. <https://www.strategicmarketresearch.com/market-report/social-commerce-market>.
- [47] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. 2019. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proc. KDD*.
- [48] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *Proc. SIGMOD*. 1539–1554.
- [49] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence Maximization: Near-Optimal Time Complexity Meets Practical Efficiency. arXiv:1404.0900 [cs.SI]
- [50] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. 2021. Multi-FR: A Multi-Objective Optimization Method for Achieving Two-sided Fairness in E-commerce Recommendation.
- [51] Haolun Wu, Bhaskar Mitra, Chen Ma, Fernando Diaz, and Xue Liu. 2022. Joint Multisided Exposure Fairness for Recommendation. In *Proc. SIGIR*.
- [52] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. TFROM: A Two-Sided Fairness-Aware Recommendation Model for Both Customers and Providers. In *Proc. SIGIR*. 1013–1022.
- [53] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In *Proc. RecSys* (Como, Italy) (RecSys '17). 107–115.
- [54] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proc. CIKM*. 529–538.
- [55] Huance Xu, Chao Huang, Yong Xu, Lianghao Xia, Hao Xing, and Dawei Yin. 2020. Global context enhanced social recommendation with hierarchical graph neural networks. In *Proc. ICDM*. IEEE, 701–710.

APPENDIX

A Non-Existence Proofs

THEOREM 1. *EQ1 allocation may not exist in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).*

PROOF. Suppose there are 5 re-sellers and 5 products. We want to allocate 3 products to each re-seller and provide at least 3 copies of recommendations for each product, i.e., $L_1 = L_2 = R_1 = R_2 = 3$. The utility values ($W_{i,j}$) are given in Table D. In such a setting, these conditions must satisfy:

- (1) Out of first 4 rows, at least 1 row will have valuation $\alpha \times 3/4$.
- (2) If we remove one item from allocation of u_5 , its utility will be 4. Now, EQ1 fails if $4 > \alpha \times 3/4 \Rightarrow \alpha < 16/3$. \square

	p_1	p_2	p_3	p_4	p_5
u_1	$\alpha/4$	$\alpha/4$	$\alpha/4$	$\alpha/4$	$10-\alpha$
u_2	$\alpha/4$	$\alpha/4$	$\alpha/4$	$\alpha/4$	$10-\alpha$
u_3	$\alpha/4$	$\alpha/4$	$\alpha/4$	$\alpha/4$	$10-\alpha$
u_4	$\alpha/4$	$\alpha/4$	$\alpha/4$	$\alpha/4$	$10-\alpha$
u_5	2	2	2	2	2

Table D

	p_1	p_2	p_3	p_4	p_5
u_1	3	3	2	1	1
u_2	3	3	2	1	1
u_3	3	3	2	1	1
u_4	3	3	2	1	1
u_5	1	1	2	3	3

Table E

Proposition 2. *Some permutation of re-sellers in round robin allocation may not find feasible EF1 solution in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).*

PROOF. Suppose there are 5 re-sellers and 5 products. We want to allocate 3 products to each re-seller and provide at least 3 copies of recommendation to each product. The utility values ($W_{i,j}$) are given in Table E. In such a setting, we can not find a feasible EF1 solution if we take any permutation that does not start with u_5 . \square

THEOREM 3. *Optimising Nash social welfare (Eq. 7) may not give EF1 solution in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5).*

PROOF. Consider an instance with two re-sellers u_1 and u_2 and four products p_1, p_2, p_3 and p_4 . Suppose each re-seller can receive exactly two products, and suppose each product can be assigned to exactly one re-seller, i.e., $L_1 = L_2 = 2$ in RE-SELLER-CONSTRAINT and $R_1 = R_2 = 1$ in PRODUCT-CONSTRAINT.

Let the utilities for re-seller u_1 be 1, 1, $2+\epsilon$, $2+\epsilon$ for p_1, p_2, p_3, p_4 respectively, where $\epsilon > 0$ is sufficiently small (say less than 0.4). Similarly, let re-seller u_2 value p_1, p_2, p_3, p_4 at $\epsilon, \epsilon, 3, 3$, respectively. A cardinality-constrained Nash optimal allocation gives p_1, p_2 to re-seller u_1 and p_3, p_4 to re-seller u_2 . This violates EF1 w.r.t. re-seller u_1 . \square

B Hardness Proofs

THEOREM 2. *Determining the existence of an EQ1 allocation in the presence of RE-SELLER-CONSTRAINT (Eq.4) and PRODUCT-CONSTRAINT (Eq.5) is NP-complete.*

PROOF. We will show a reduction from the 3-Partition problem, which is known to be NP-complete [26].

Definition 5 (3-Partition problem). *An instance of the 3-Partition problem involves a set of $3m$ numbers b_1, \dots, b_{3m} that sum up to mT . The goal is to partition the numbers into m triplets with equal sums. The problem remains NP-complete even if all b_i 's are close to $T/3$.*

Fair division instance to find EQ1 under cardinality constraints: Construct a fair division instance with a set of $m+1$ re-sellers and $3m+3$ items. There are m main re-sellers and one dummy re-seller. The items consist of $3m$ partition items g_1, \dots, g_{3m} and three dummy items. Each main re-seller values each partition item g_i at b_i , and each dummy item at 0. The dummy re-seller values every item at $T/2$. The cardinality constraints require that each re-seller receives exactly three items, and one copy of the items is allocated.

In the forward direction, given a solution to 3-Partition, one can construct a feasible EQ1 allocation by assigning the partition items among the main re-sellers in accordance with the solution to 3-Partition, and give the three dummy items to the dummy re-seller. **In the reverse direction,** suppose A is a feasible EQ1 allocation. Then, by the cardinality constraint, the dummy re-seller must receive three items under A , each of which it values at $T/2$. Therefore, to satisfy EQ1, every main re-seller must have a value of at least T . This means that each main re-seller must get three partition items (recall that each b_i is close to $T/3$, so any two b_i 's together cannot take us to T). This, in turn, implies that all dummy items must go to the dummy re-seller. Therefore, under A , we must allocate partition items so that each main re-seller gets exactly three partition items, and its utility is at least T . Since the partition items have a total value of mT , each main re-seller must have a value of exactly T , giving us a solution of 3-Partition. Hence, finding EQ1 allocation under cardinality constraints is NP-complete. \square

THEOREM 4. *Two-sided cardinality constrained Nash Optimization (2S-CARDNASHOPT) problem is NP-hard.*

PROOF. (sketch) We briefly outline a reduction from a balanced version of the Partition problem [43]. The input consists of a set of $2r$ numbers a_1, \dots, a_{2r} that sum up to $2T$. The goal is to partition these numbers into two sets of cardinality r each such that the numbers in each set sum to T .

The reduced fair division instance is constructed as follows: There are $2r$ items/products p_1, \dots, p_{2r} . Each item can be allocated to exactly one reseller (i.e., $R_1 = R_2 = 1$). Each reseller can get exactly r items (i.e., $L_1 = L_2 = r$). The decision threshold for Nash welfare is T .

The implication in forward direction is easy to see. In the reverse direction, we observe that the arithmetic mean of the value is exactly T and the geometric mean is at least T . Thus, by the AM-GM inequality, the utilities of resellers must be equal, inducing the desired partition. \square

C Complexity Analysis

Suppose there are m re-sellers and n products. The minimum and maximum size of allocation list for each re-seller is L_1 and L_2 respectively. R_1 and R_2 are the minimum and maximum number of assigned re-sellers for each product respectively.

C.1 The complexity of GREEDYNASH algorithm. The complexity of algorithm 1 (GREEDYNASH) is $O(mn + R_1^2 L_2^3)$. The complexity of steps (1-9 and 15-18) is $O(mn)$. The greedy replacement procedure (steps 10-14) is bounded with $O(R_1^2 L_2^3)$ for $R_1 = \lfloor \alpha \times L_1 \times m/n \rfloor$, where $0 < \alpha \leq 1$. At step 7, a product (p) is not allocated to any re-seller, when at most R_1 re-sellers are assigned products less than L_2 . Note that these re-sellers are already in allocation of (p), say such re-sellers U_R . This case of not finding feasible re-seller can arise for at most L_2 products, say those products P_L . The re-sellers U_R can be assigned with at most $R_1 \times L_2$ unique products whose cardinality can exceed R_1 (at step 9). So there can be at most $R_1 \times L_2 \times L_2$ re-sellers which are assigned products whose cardinality exceed R_1 , say such re-sellers $U_{RL_2^2}$. To balance the cardinality of products P_L , at most $R_1 \times L_2$ assignments are done by greedily replacing $R_1 \times L_2$ products across $R_1 \times L_2 \times L_2$ re-sellers of $U_{RL_2^2}$. So complexity of replacement procedure (steps 10-14) is bounded with $O(R_1^2 L_2^3)$.

C.2 The complexity of SEAL algorithm. The complexity of algorithm 2 (SEAL) is $O(mn \log m + L_2^2 R_1^2)$. The complexity of steps (1-4) is $O(m \log m)$, steps (5-9 and 17-20) is $(mn \log m)$, and steps (10-11 and 21-22) is $O(m \log m)$. The complexity of steps (12-16) is bounded by $O(L_2^2 R_1^2)$ for $R_1 = \lfloor \alpha \times L_2 \times m/n \rfloor$, where $0 < \alpha \leq 1$. A re-seller can not find feasible products at step 8, if there are at most L_2 products which are allocated to less than R_1 re-sellers. Say such products P_L . There can be at most $L_2 \times R_1$ re-sellers which are allocated to products whose cardinality exceed R_1 , say such re-sellers U_{LR} . To balance the cardinality of products P_L , at most $R_1 \times L_2$ assignments are done by greedily replacing $R_1 \times L_2$ products across $R_1 \times L_2$ re-sellers of U_{LR} . So complexity of steps (12-16) is $O(L_2^2 R_1^2)$.

D Comparing GREEDYNASH and SEAL

	p_1	p_2	p_3
u_1	7	1	2
u_2	5.5	2	2.5
u_3	5	4	1

Table F: GREEDYNASH is better than SEAL

	p_1	p_2	p_3
u_1	7	1	2
u_2	6	1.5	2.5
u_3	5	4	1

Table G: SEAL is better than GREEDYNASH

While comparing SEAL and GREEDYNASH, either one might output allocations with better Nash Welfare. Lets demonstrate it with an example. Suppose there are three re-sellers and three products. We want to allocate 2 products to each re-seller and provide 2 copies of recommendations for each product, i.e., $L_1 = L_2 = R_1 = R_2 = 2$. The utility values ($W_{i,j}$) are given in Tables F and G. Suppose we allocate the first product to users in order u_1 , u_2 , and u_3 , and we choose product ordering p_1 , p_2 , and p_3 for allocation. In case of Table F GREEDYNASH provide allocations ($u_1[p_1, p_3]$, $u_2[p_1, p_2]$, $u_3[p_2, p_3]$) with $Nash = 337.5$, while SEAL provides allocations ($u_1[p_1, p_2]$, $u_2[p_1, p_3]$, $u_3[p_2, p_3]$) with $Nash = 320$. In cases of Table G, GREEDYNASH provides allocations ($u_1[p_1, p_3]$, $u_2[p_1, p_2]$, $u_3[p_2, p_3]$) with $Nash = 337.5$, while SEAL provides allocations ($u_1[p_1, p_2]$, $u_2[p_1, p_3]$, $u_3[p_2, p_3]$) with $Nash = 340$.

Metric	SEAL	GREEDYNASH	LPT	UNCONSGREEDYNASH
Revenue (in USD)	310.57k	309.4k	306.35k	308.5k
Avg. gini (users)	0.19	0.21	0.07	0.09
Avg. income gap	398.33	1080.40	342.94	346.68
5th %tile of products allocated	15	15	1	1
95th %tile of products allocated	15	15	19	26
Variance in products allocated	0	0	30.95	53.81

Table H: Comparison of SEAL, GREEDYNASH, LPT, and UNCONSGREEDYNASH on real social commerce dataset to show the need of re-seller side constraints

E Importance of RE-SELLER-CONSTRAINT (Eq.4)

RE-SELLER-CONSTRAINT constraint is a practical necessity. Social-commerce platforms work within the design constraints of a UI, which typically shows a certain number of products. length of the recommendation list for each re-seller is controlled through re-seller-side constraints. To verify this empirically, we benchmark our algorithms against two adapted baselines without RE-SELLER-CONSTRAINT.

- LPT [40] is an extension of Graham(1966)'s classical "Longest-processing-time-first(LPT)" algorithm for job scheduling. We adapt it to showcase the use of re-seller side cardinality constraints while optimizing Nash. In our adaptation, we iteratively assign products to the least-happy reseller while ensuring that product-side cardinality constraints are satisfied.
- UNCONSGREEDYNASH is un-constrained version of GREEDYNASH where RE-SELLER-CONSTRAINT (Eq.4) is not enforced.

We set $R_2 = R_1 + 1$ for comparing LPT with GREEDYNASH and SEAL. SEAL and GREEDYNASH outperforms LPT and UNCONSGREEDYNASH (see Table H). Specifically, when UNCONSGREEDYNASH optimizes Nash, the number of products allocated to a reseller varies widely, violating re-seller side constraints. For example, 15.35% resellers got less than 5 products, whereas 25.82% resellers got more than 20 products. A Similar trend is observed for the LPT too. Since selling success depends on multiple factors such as catalog diversity, finding interested customers, etc., such variability hurts the re-seller experience.

F Synthetic Dataset Generation

To compare the proposed heuristic to the optimal allocation, we generate 100 synthetic datasets with each containing 100 users and 100 products. Each product's (revenue) is set to an integer chosen uniformly at random in the range $[1, 1000]$. Note that Nash optimization is free from scaling the utilities of users, and the need for integer valuation for MILP is mentioned in § 4.1. Each dimension (product) in the expertise vector of an user is set to a value chosen uniformly at random from the range $[0, 1]$.