

Fair Distribution of Delivery Orders

Hadi Hosseini¹, Shivika Narang² and Tomasz Was³

¹Pennsylvania State University

²University of New South Wales

³University of Oxford

hadi@psu.edu, s.narang@unsw.edu.au, tomasz.was@cs.ox.ac.uk

Abstract

We initiate the study of fair distribution of delivery tasks among a set of agents wherein delivery jobs are placed along the vertices of a graph. Our goal is to fairly distribute delivery costs (modeled as a submodular function) among a fixed set of agents while satisfying some desirable notions of economic efficiency. We adopt well-established fairness concepts—such as *envy-freeness up to one item* (EF1) and *minimax share* (MMS)—to our setting and show that fairness is often incompatible with the efficiency notion of *social optimality*. Yet, we characterize instances that admit fair and socially optimal solutions by exploiting graph structures. We further show that achieving fairness along with Pareto optimality is computationally intractable. Nonetheless, we design an XP algorithm (parameterized by the number of agents) for finding MMS and Pareto optimal solutions on every tree instance, and show that the same algorithm can be modified to find efficient solutions along with EF1, when such solutions exist. We complement these results by theoretically and experimentally analyzing the price of fairness.

1 Introduction

With the rise of digital marketplaces and the gig economy, package delivery services have become crucial components of e-commerce platforms like Amazon, AliExpress, and eBay. In addition to these novel platforms, traditional postal and courier services also require swift turnarounds for distributing packages. Prior work has extensively investigated the optimal partitioning of tasks among the delivery agents under the guise of *vehicle routing* problems (see [Toth and Vigo, 2002] for an overview). However, these solutions are primarily focused on optimizing the efficiency (often measured by delivery time or distance travelled [Kleinberg *et al.*, 1999; Pioro, 2007]), and do not consider fairness towards the delivery agents. This is particularly important in the settings where agents do not receive monetary compensation, e.g., in volunteer-based social programs such as Meals on Wheels [O’Dwyer and Timonen, 2009].

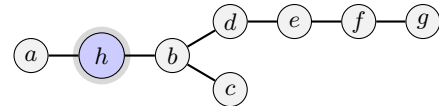


Figure 1: An example graph with the hub, h , marked.

We consider fair distribution of delivery orders that are located on the vertices of a connected graph, containing a warehouse (the *hub*). Agents are tasked with picking up delivery packages (or *items*) from the fixed hub, deliver them to the vertices, and return to the hub. In this setting, the cost incurred by an agent i is the total distance traveled, that is, the total number of the edges traversed by i in the graph. Let us illustrate this through an example.

Example 1. Consider seven delivery orders $\{a, b, \dots, g\}$ and a hub (h) that are located on a graph as depicted in Figure 1. An agent’s cost depends on the graph structure and is submodular. For instance, the cost of delivering an order to vertex f is the distance from the hub h to f , which is 4¹; but the cost of delivering to f and g is only 5 since they can both be serviced in the same trip.

Let there be two (delivery) agents. If the objective were to simply minimize the total distance travelled (social optimality), then there are two solutions with the total cost of 7: either one agent delivers all the items or one agent services a while the other services the rest. However, these solutions do not distribute the delivery orders fairly among the agents.

One plausible fair solution may assign $\{a, b, f\}$ to the first agent and $\{c, d, e, g\}$ to the other, minimizing the cost discrepancy. However, since both agents benefit from exchanging f for c , this allocation is not efficient or, more precisely, it is not Pareto optimal. After the exchange, the first agent services $\{a, b, c\}$ and the second agent $\{d, e, f, g\}$, which in fact is a Pareto optimal allocation.

The above example captures the challenges in satisfying fairness in conjunction with efficiency, and consequently, motivates the study of fair distribution of delivery orders. The literature on fair division has long been concerned with the fair allocation of goods (or resources) [Lipton *et al.*, 2004;

¹Formally, there is also the cost of returning to h , but since, on trees, each edge must be traversed by an agent twice (once in each direction), we do not count the return cost for simplicity.

Budish, 2011; Barman *et al.*, 2019; Freeman *et al.*, 2019], chores (or tasks) [Chen and Liu, 2020; Ebadian *et al.*, 2022; Huang and Lu, 2021; Hosseini *et al.*, 2022], and mixtures thereof [Bhaskar *et al.*, 2021; Aziz *et al.*, 2022; Caragiannis and Narang, 2023; Hosseini *et al.*, 2023b]. It has resulted in a variety of fairness concepts and their relaxations. Most notably, *envy-freeness* and its relaxation—*envy-freeness up to one item* (EF1) [Lipton *et al.*, 2004]—have been widely studied in the context of fair division. Another well-studied fairness notion, minimax share (MMS) [Budish, 2011], requires that agents receive cost no more than what they would have received if they were to create (almost) equal partitions. A key question is how to adopt these fairness concepts to the delivery problems, and whether these fairness concepts are compatible with natural efficiency requirements.

1.1 Contributions

We initiate the study of fair distribution of delivery tasks among a set of agents. The tasks are placed on the vertices of an acyclic or tree graph and have submodular service costs. The primary objective is to find a fair partition of m delivery orders (represented by vertices of a graph), starting from a fixed hub, among n agents. We consider two well-established fairness concepts of EF1 and MMS and explore their existence and computation along with efficiency notions of social optimality (SO) and Pareto optimality (PO).

Table 1 summarizes our results on the coexistence of fairness and efficiency in arbitrary graphs. We first show that an EF1 allocation always exists and can be computed in polynomial time on trees (Proposition 1). In contrast, an MMS allocation is guaranteed to exist, but its computation remains NP-hard (Theorem 1). Of the combinations of fairness and efficiency, only MMS and PO allocations are guaranteed to exist. Finding a fair and efficient allocation remains NP-hard.

We note that our intractability results hold even for the restricted case of unweighted tree graphs. Thus, while our existence and hardness results hold for cyclic and/or weighted graphs (see the full version of the paper [Hosseini *et al.*, 2023a] for details on how our results extend beyond unweighted trees), in order to establish a solid baseline for further work on fair delivery, in this paper we focus on unweighted tree instances.

Indeed, a popular technique to deal with computationally difficult problems on graphs is to provide algorithms with complexity parameterized by some measure of tree-resemblance [Cygan *et al.*, 2015]. Trees can be also motivated in practice. Many suburban neighborhoods in the United States, among other countries, are designed with a cul-de-sac layout, with houses/mailboxes typically located at uniform distances. Then, a group of agents distributing a newspaper or a leaflet to every house in such neighborhood, will face an unweighted tree instance of our problem.

For this setting, we characterize the conditions for the existence of fair (EF1 and MMS) and efficient (PO and SO) allocations in tree instances. In particular, we find a necessary condition for an allocation to be EF1 and PO (Proposition 5), which helps us prove that such an allocation must be leximin optimal. This leads us to an intriguing result that an EF1 and PO allocation always satisfies MMS as well (Theorem 3). We

		–	PO	SO
EF1	existence	✓	✗ (Prop. 5a)	✗ (Thm. 2)
	computation	P (Prop. 1)	NP-h (Prop. 7)	NP-h (Prop. 3)
MMS	existence	✓	✓ (Prop. 5b)	✗ (Thm. 2)
	computation	NP-h (Thm. 1)	NP-h (Prop. 7)	NP-h (Prop. 3)

Table 1: The summary of our results on EF1 and MMS in conjunction with efficiency requirements of PO and SO. ✓ denotes that the allocation always exists, and ✗ that it may not exist. We note that every NP-hard problem here can be solved with an XP algorithm parameterized by the number of agents (Thm. 5).

use these observations to circumvent our hardness results by designing an XP algorithm (Algorithm 1), parameterized by the number of agents, that allows us to find an MMS and PO solution and check whether an instance admits an EF1 and PO allocation as well (Theorem 5).

Subsequently, we study the price of fairness in our setting by establishing its worst case bounds and typical values in a theoretical and experimental analysis, respectively.

All missing proofs can be found in the full version of the paper [Hosseini *et al.*, 2023a]. There, we also present additional results on a stronger fairness notion of envy-freeness up to any item (EFX). We note that an EFX allocation always exists in this setting, however EFX combined with some efficiency requirement, for large classes of graphs, becomes as restrictive as envy-freeness. We also discuss how our results extend to graphs with weighted edges and/or cycles and provide additional experiments.

1.2 Related Work

Fair division of indivisible items has garnered much attention in recent years. Several notions of fairness have been explored in this space, with EF1 [Lipton *et al.*, 2004; Budish, 2011; Barman *et al.*, 2019; Caragiannis and Narang, 2023] and MMS [Ghodsi *et al.*, 2018; Barman and Krishnamurthy, 2020; Hosseini and Searns, 2021] being among the most prominent ones. An important result is from Caragiannis *et al.* [2019] showing that an EF1 and PO allocation is guaranteed to exist for items with non-negative additive valuations. Some prior work has also looked at fair division on graphs [Bouveret *et al.*, 2017; Truszczynski and Lonc, 2020; Misra *et al.*, 2021; Bilò *et al.*, 2022; Misra and Nayak, 2022], but in the settings that are very different from assigning delivery orders. The majority of these works identified the vertices of a graph with goods and analyzed how to fairly partition the graph into contiguous pieces.

Recent works have explored fairness in delivery settings [Gupta *et al.*, 2022; Nair *et al.*, 2022] or ride-hailing platforms [Esmaili *et al.*, 2023; Sánchez *et al.*, 2022]. However, these studies are mainly experimental and do not provide any positive theoretical guarantees. While fairness has been studied in routing problems, the aim has been to balance the amount of traffic on each edge [Kleinberg *et al.*, 1999; Pioro, 2007], which does not capture the type of delivery instances that we investigate in this paper. In the full version of the paper [Hosseini *et al.*, 2023a], we provide an extended review of the literature.

2 Our Model

We denote a *delivery instance* by an ordered triple $I = \langle [n], G, h \rangle$, where $[n]$ is a set of agents, $G = (V, E)$ is an undirected acyclic graph (or a tree) of delivery orders rooted in $h \in V$. The special vertex, i.e., the root, h is called the *hub*. The goal is to assign each vertex in graph G , except for the hub, to a unique agent that will *service* it. Formally, an allocation $A = (A_1, \dots, A_n)$ is an n -partition of vertices in $V \setminus \{h\}$. We are interested in *complete* allocations such that $\cup_{i \in [n]} A_i = V \setminus \{h\}$, and denote the set of all complete allocations by Π^n .

An agent's *cost* for servicing a vertex $v \in V$, denoted by $c(v)$, is the length of the shortest path from the hub h to v . An agent's cost for servicing a set of vertices $S \subseteq V \setminus \{h\}$ is equal to the minimum length of a walk that starts and ends in h and contains all vertices in S divided by two.² A walk to serve vertices in S may pass through vertices in some superset of S , i.e., $S' \supseteq S$. Thus, the cost function c is *submodular* and belongs to the class of *coverage functions*. We use $G|_S$ to denote the minimal connected subgraph containing all vertices in $S \cup \{h\}$. Thus, we have $c(S) = |E(G|_S)|$. We say that an agent servicing S , *visits* all vertices in $G|_S$.

Fairness Concepts. The most plausible fairness notion is *envy-freeness* (EF), which requires that no agent (strictly) prefers the allocation of another agent. An EF allocation may not exist; consider one delivery order and two agents. A prominent relaxation of EF is *envy-freeness up to one order* (EF1) [Lipton *et al.*, 2004; Budish, 2011], which requires that every pairwise envy can be eliminated by a removal of a single order served by the envious agent.

Definition 1 (Envy-Freeness up to One Order (EF1)). *An allocation A is EF1 if for every pair $i, j \in [n]$, either $A_i = \emptyset$ or there exists $x \in A_i$ such that $c(A_i \setminus \{x\}) \leq c(A_j)$.*

Another well-studied notion is *minimax share* (MMS), which ensures that each agent gets at most as much cost as they would if they were to create an n -partition of the delivery orders but then receive their least preferred bundle. This notion is an adaptation of maximin share fairness—that deals with positive valuations [Budish, 2011]—to the settings that deals with negative valuations and has been recently studied in fair allocation of chores (see e.g., Huang and Lu [2021]).

Definition 2 (Minimax Share (MMS)). *An agent's minimax share cost is given by*

$$\text{MMS}_i(I) = \min_{A \in \Pi^n} \max_{i \in [n]} c(A_i),$$

An allocation A is MMS if $c(A_i) \leq \text{MMS}_i(I)$ for all $i \in [n]$.

Under additive non-negative identical valuations, MMS implies EF1 (for completeness, we give a formal proof in the full version of the paper). However, since in our case the costs are submodular, EF1 and MMS do not imply each other, even though the cost functions are identical (see Example 2).

²On trees, in each such walk, each edge is traversed by an agent two times (once in both directions). For simplicity, we drop the return cost, hence the division by 2.

Economic Efficiency. Our first notion of efficiency is social optimality that requires that the aggregate cost of the agents is minimum, i.e., equal to the number of edges in the graph. In such a case, each vertex (except for the hub) is visited by only one agent.

Definition 3 (Social Optimality (SO)). *An allocation A is socially optimal if $\sum_{i=1}^n c(A_i) = |E(G)|$. In other words, for every pair of agents $i \neq j \in [n]$, the only vertex they both visit is the hub, i.e., $V(G|_{A_i}) \cap V(G|_{A_j}) = \{h\}$.*

Here, $V(\cdot)$ takes as an input a subgraph and returns the set of vertices in it. An allocation that assigns all vertices to a single agent is vacuously SO. However, as we discussed in Example 1 it may result in a very unfair distribution of orders. Therefore, we consider a weaker efficiency notion that allows for some overlap in vertices visited by the agents.

Definition 4 (Pareto Optimality (PO)). *An allocation A Pareto dominates A' if $c(A_i) \leq c(A'_i)$, for every agent $i \in [n]$, and there exists some agent $j \in [n]$ such that $c(A_j) < c(A'_j)$. An allocation is Pareto optimal if it is not Pareto dominated by any other allocation.*

In other words, an allocation is PO if we cannot reduce the cost of one agent without increasing it for some other agent. Let us now follow up on Example 1 and analyze allocations satisfying our notions.

Example 2. *Consider the instance with 2 agents and the graph from Figure 1. As previously noted, there are only two SO allocations and neither is EF1 or MMS.*

PO allocation $(\{d, e, f, g\}, \{a, b, c\})$ satisfies MMS (vertex g must be serviced by some agent, hence the MMS cost cannot be smaller than 5), but it is not EF1. In fact, there is no EF1 and PO allocation in this instance, as an agent servicing g has to service f, e and d as well (otherwise giving them to this agent would be a Pareto improvement). But then, even when we assign the remaining vertices, a, b, c , to the second agent, the allocation would violate EF1.

Finally, observe that allocation $(\{a, b, f\}, \{c, d, e, g\})$ is EF1, but not MMS.

In order to study the (co-)existence of these fairness and efficiency notions, we use leximin optimality. Let us first set up some notation before defining it. Given an allocation A , we can sort it in non-increasing cost order to obtain allocation $B = \text{sort}(A)$ such that $c(B_1) \geq c(B_2) \geq \dots \geq c(B_n)$ and $B_i = A_{\pi(i)}$ for every agent $i \in [n]$ and some permutation of agents π .

Definition 5 (Leximin Optimality). *An allocation A leximin dominates an allocation A' if there is agent $i \in [n]$ such that $c(B_i) < c(B'_i)$ and $c(B_j) = c(B'_j)$ for every $j \in [i - 1]$, where $B = \text{sort}(A)$ and $B' = \text{sort}(A')$. An allocation is leximin optimal if it is not leximin dominated by any other allocation.*

In other words, a leximin optimal allocation first minimizes the cost of the worst-off agent, then minimizes the cost of the second worst-off agent, and so on. We note that a leximin optimal allocation is always Pareto optimal as well.

3 Existence of Fair Allocations

In this section, we consider the existence and computation of EF1 and MMS allocations.

Recall that our cost functions are submodular and *monotone*. Thus, an EF1 allocation can be obtained by adapting an *envy-graph* algorithm from Lipton *et al.* [2004]: start with each agent having an empty set, pick an agent who currently has minimum cost (i.e., a sink of the envy-graph) and out of the unassigned vertices give them the one that results in a minimal increase of the cost. Repeat this till all vertices in $V \setminus \{h\}$ are allocated. This algorithm always returns an EF1 solution as long as valuations are identical and monotone.

Proposition 1. *Given a delivery instance $I = \langle [n], G, h \rangle$, an EF1 allocation always exists and can be computed in polynomial time.*

An MMS allocation in our setting always exists. It follows from the fact that agents have identical cost functions (an allocation that minimizes the maximum cost will satisfy MMS). However, finding such an allocation is NP-hard. To establish this, we first show the hardness of finding the MMS cost.

Proposition 2. *Given a delivery instance $I = \langle [n], G, h \rangle$, finding the MMS cost is NP-hard.*

Theorem 1. *Given a delivery instance $I = \langle [n], G, h \rangle$, an MMS allocation always exists, however, finding such an allocation is NP-hard.*

Proof. The existence of MMS allocations follows from the cost functions being identical. The same allocation will give the MMS threshold for all agents, hence would satisfy MMS for all. Further, for each instance I and MMS allocation A the cost of the agent that is the worst off, must be equal to MMS cost, i.e., $\max_{i \in [n]} c(A_i) = \text{MMS}_i(I)$. Hence, if we had a polynomial time algorithm for finding an MMS allocation, we would be able to find MMS cost by looking at the maximal cost of an agent. Since by Proposition 2 the latter is NP-hard, there is no such algorithm unless $P = NP$. \square

4 Characterizing Fair and Efficient Solutions

Example 2 established the possible incompatibility of fairness and efficiency in our setting. In this section, we exploit the structure provided by trees to characterize the space of delivery instances for which fair and efficient allocations exist. We first discuss social optimality and then turn our attention to Pareto optimality.

4.1 Social Optimality

In every socially optimal allocation, each edge is traversed by a unique agent, thus the sum of all agents costs is m (the number of edges in a tree with $m + 1$ vertices). This is a very demanding condition, hence it is not surprising that it may be impossible to satisfy it together with some fairness requirement. We show that checking if there exists a social optimal and fair allocation is computationally hard.

Proposition 3. *Given a delivery instance $I = \langle [n], G, h \rangle$, an SO allocation that satisfies EF1 or MMS need not exist. Moreover, checking whether an instance admits such an allocation is NP-hard.*

Despite the computational hardness result, we exploit the tree structure to characterize instances with such allocations.

Theorem 2. *Given a delivery instance $I = \langle [n], G, h \rangle$:*

- An EF1 and SO allocation exists if and only if there is a partition (P_1, \dots, P_n) of branches out of h such that $\sum_{B \in P_i} |B| - \sum_{B \in P_j} |B| \leq 1$, for every $i, j \in [n]$,*
- An MMS and SO allocation exists if and only if there is a partition (P_1, \dots, P_n) of branches outgoing from h such that $\sum_{B \in P_i} |B| \leq \text{MMS}_i(I)$ for every $i \in [n]$.*

By Proposition 3, checking the condition in point *a* of the above theorem is NP-hard. However, we develop a polynomial-time verifiable necessary condition for EF1 and SO existence using the notion of the *center* of the graph (which was studied in computational social choice [Skibski, 2023] and in theoretical computer science in general [Goldman, 1971]).

Definition 6. *The center of a tree $G = (V, E)$ is a set of vertices $C = \arg \min_{v \in V} \sum_{u \in V} \text{dist}(u, v)$, where $\text{dist}(x, y)$ is the length of a shortest path from x to y .*

Proposition 4. *Given a delivery instance $I = \langle [n], G, h \rangle$, there exists an EF1 and SO allocation only if the hub is in the center of the tree.*

4.2 Pareto Optimality

Given the challenges in satisfying SO, we now focus on Pareto optimality. Recall that all SO allocations are PO, but not vice versa. We begin by noting that an MMS and PO allocation always exists, but the same is not true for an EF1 and PO allocation.

Proposition 5. *Given a delivery instance $I = \langle [n], G, h \rangle$,*

- an EF1 and PO allocation need not exist,*
- an MMS and PO allocation always exists.*

The central result of this section is the proof that every EF1 and PO allocation will satisfy MMS as well. This comes in contrast to typical fair chore division settings, where under additive preferences EF1 and MMS are independent notions even in the presence of efficiency requirements. To this end, we first prove an insightful necessary condition for EF1 and PO allocations: in every such allocation, the pairwise difference in the costs of agents cannot be greater than 1.

Proposition 6. *Given a delivery instance $I = \langle [n], G, h \rangle$ and an EF1 allocation A , if $|c(A_i) - c(A_j)| > 1$ for some agents $i, j \in [n]$, then A is not PO.*

Proof. Without loss of generality, assume that A is sorted in non-increasing cost order, i.e., $c(A_1) \geq \dots \geq c(A_n)$ (otherwise we can relabel the agents). We will show that if $c(A_n) < c(A_1) - 1$, then A is not PO, i.e., it is Pareto dominated by some allocation A' (not necessarily EF1).

For every vertex $x \in V \setminus \{h\}$, by $p(x)$ let us denote the *parent* of x in a tree G rooted in h . Also, for every agent $i \in [n]$, let $w(i)$ be the *worst* vertex in i 's bundle, i.e., the vertex which on removal gives the largest decrease in cost (if there is more than one we take an arbitrary one). Formally, $w(i) = \arg \max_{x \in A_i} c(A_i) - c(A_i \setminus \{x\})$. Since A is EF1,

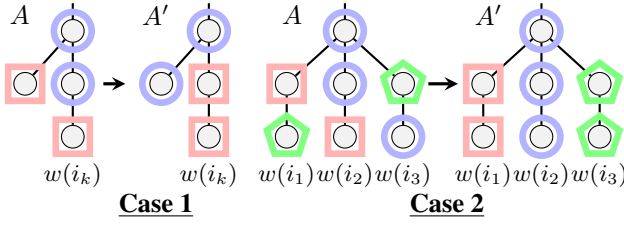


Figure 2: Illustrating the proof of Prop. 6. In Case 1, i_k (red squares) exchanges its bundle with i^* (blue circles) except for $w(i_k)$. In Case 2, i_1, i_2 , and i_3 (pentagons, squares, and circles, resp.) swap their worst vertices along the cycle.

for every agent i with maximal cost, i.e., such that $c(A_i) = c(A_1)$, we have

$$c(A_i \setminus \{w(i)\}) \leq c(A_n) < c(A_i) - 1. \quad (1)$$

Observe that this is only possible if the parent of $w(i)$ is not serviced by i , i.e., $p(w(i)) \notin A_i$. To construct allocation A' which Pareto dominates A , we look at the agent servicing the parent of the worst vertex of agent 1, call this agent i_1 . If i_1 incurs maximum cost, we look at the agent servicing the parent of the worst vertex of i_1 . We continue in this manner and obtain a maximal sequence of pairwise disjoint agents $1 = i_0, i_1, \dots, i_k$ such that $p(w(i_{s-1})) \in A_{i_s}$ and $c(A_{i_s}) = c(A_1)$ for every $s \in [k]$. The cost incurred by the agent servicing the parent of the worst vertex of i_k , which we denote by i^* (i.e., $p(w(i_k)) \in A_{i^*}$) can create two cases. Either i^* does not incur maximum cost, i.e., $c(A_{i^*}) < c(A_1)$ (Case 1), or it already appears in the sequence, i.e., $i^* = i_j$ for some $j < k$ (Case 2).

Case 1. Consider allocation A' obtained from A by exchanging the bundles of agent i_k and agent i^* with the exception of $w(i_k)$ (which continues to be serviced by i_k). See Figure 2 for an illustration. Formally, $A'_{i_k} = A_{i_k} \setminus \{w(i_k)\}$, $A'_{i^*} = A_{i^*} \cup \{w(i_k)\}$, and $A'_t = A_t$, for every $t \in [n] \setminus \{i^*, i_k\}$. Since costs of agents in $[n] \setminus \{i^*, i_k\}$ are not affected, it suffices to prove that the cost of either i_k or i^* decreases without increasing the other's cost. To this end, observe that since parent of $w(i_k)$ belongs to A_{i^*} , adding this vertex to A_{i^*} increases the cost by 1, i.e.,

$$c(A'_{i_k}) = c(A_{i^*}) + 1. \quad (2)$$

Now, let us consider two subcases based on the original difference in costs of agents i_k and i^* .

If it is greater than one, i.e., $c(A_{i_k}) > c(A_{i^*}) + 1$, then from Equation (2) we get that $c(A'_{i_k}) = c(A_{i^*}) + 1 < c(A_{i_k})$. Hence, the cost of i_k decreases. On the other hand, by Equation (1), $c(A'_{i^*}) = c(A_{i_k} \setminus \{w(i_k)\}) \leq c(A_n) \leq c(A'_{i^*})$, so agent i^* does not suffer from the exchange.

Otherwise, the difference in costs is exactly one, i.e., $c(A_{i_k}) = c(A_{i^*}) + 1$. Now, the cost of i_k stays the same by Equation (2). However, since $c(A_n) < c(A_1) - 1$, it must be that $c(A_n) < c(A_{i^*})$. Thus, from Equation (1) we have $c(A'_{i^*}) \leq c(A_n) < c(A_{i^*})$, i.e., the cost of i^* decreases.

Case 2. When $i^* = i_j$ for some $j < k$, we have a cycle of agents $i^* = i_j, i_{j+1}, \dots, i_k$ such that $c(A_{i_s}) = c(A_1)$ and

$p(w(i_s)) \in A_{i_{s+1}}$ for every $s \in \{j, \dots, k\}$ (we denote i_j as i_{k+1} as well for notational convenience). Here, we consider two subcases, based on whether it happens somewhere in the cycle that the parent of the worst vertex of one agent is the worst vertex of the next agent, i.e., $p(w(i_s)) = w(i_{s+1})$ for some $s \in \{j, \dots, k\}$.

Case 2a. If this is the case, consider A' obtained from A by giving $w(i_{s+1})$ to agent i_s . Formally, $A'_{i_{s+1}} = A_{i_{s+1}} \setminus \{w(i_{s+1})\}$, $A'_{i_s} = A_{i_s} \cup \{w(i_{s+1})\}$, and $A'_t = A_t$, for every $t \in [n] \setminus \{i_s, i_{s+1}\}$. Now, the cost of agent i_{s+1} decreases as it no longer services its worst vertex. Since agent i_s was servicing a child of $w(i_{s+1})$, it was visiting $w(i_{s+1})$ on the way. Hence, the cost of i_s stays the same. As the bundles of the remaining agents did not change, A' Pareto dominates A .

Case 2b. Finally, if in the cycle there is no agent for which the parent of its worst vertex is the worst vertex of the next agent, we swap the worst vertices along the cycle (e.g. in Figure 2). Formally, $A'_{i_s} = A_{i_s} \setminus \{w(i_s)\} \cup \{w(i_{s-1})\}$ for every $s \in \{j+1, \dots, k+1\}$ and $A'_i = A_i$ for every $i \in [n] \setminus \{i_{j+1}, \dots, i_{k+1}\}$. As each i_s is servicing the parent of the worst vertex of the previous agent in the cycle, i.e., $p(w(i_{s-1})) \in A'_{i_s}$, servicing $w(i_{s-1})$ incurs an additional cost of 1. However, from Equation (1) giving away the worst vertex decreases the cost by more than 1. Hence, the cost of each agent in the cycle decreases. The other agents' costs stay the same, thus A' Pareto dominates A . \square

We can now show that EF1 and PO imply MMS.

Theorem 3. *Given a delivery instance $I = \langle [n], G, h \rangle$, every EF1 and PO allocation satisfies MMS.*

Proof. We show a stronger result that an EF1 and PO allocation A is necessarily leximin optimal (thus, also MMS). Without loss of generality, let A be s.t. $c(A_1) \geq \dots \geq c(A_n)$. By contradiction, suppose that there exists A' (also sorted) that leximin dominates A , i.e., there exists $i \in [n]$ such that $c(A'_i) < c(A_i)$ and $c(A'_j) = c(A_j)$ for every $j \in [i-1]$. Fix an agent $j \in [n]$ such that $j > i$. From Proposition 6 we know that its cost is either $c(A_j) = c(A_i)$ or $c(A_j) = c(A_i) - 1$. On the other hand, we assumed that $c(A'_j) \leq c(A'_i) < c(A_i)$. Hence, $c(A'_j) \leq c(A_i) - 1 \leq c(A_j)$. Thus, A' also Pareto dominates A , which is a contradiction. \square

From the proof of the above theorem we obtain the following characterization.

Corollary 1. *Given a delivery instance $I = \langle [n], G, h \rangle$, an EF1 and PO allocation exists if and only if for every leximin optimal allocation A , $\max_{i,j \in [n]} |c(A_i) - c(A_j)| \leq 1$.*

Finally, by modifying the proof of Proposition 2 we show that deciding if there exists a PO allocation that is EF1 is also NP-hard. Also, we note that hardness for MMS and PO is directly implied by Proposition 2.

Proposition 7. *Given a delivery instance $I = \langle [n], G, h \rangle$, checking whether there exists an EF1 and PO or finding an MMS and PO allocation is NP-hard.*

Algorithm 1 FindParetoFrontier(n, G, h)

```

1:  $\mathcal{F} \leftarrow [(\emptyset, \dots, \emptyset)]$ 
2: for  $u \in \text{children of } h$  do
3:    $T_u \leftarrow$  a subtree rooted in  $u$ 
4:    $\mathcal{F}' \leftarrow \text{FindParetoFrontier}(n, T_u, u)$ 
5:   for  $A \in \mathcal{F}'$  do add  $u$  to  $A_1$ 
6:    $\mathcal{F} \leftarrow$  maximal set of sorted combinations of allocations from
      $\mathcal{F}$  and  $\mathcal{F}'$  such that none is weakly Pareto dominated by an-
     other within the set
7: end for
8: return  $\mathcal{F}$ 
    
```

5 Computing Fair and Efficient Solutions

We have shown that finding a fair and efficient allocation (or deciding if it exists) is computationally hard. In this section, we develop a recursive algorithm for computing each combination of the fairness-efficiency notions that is XP with respect to the number of agents, i.e., when the number of agents is bounded, the running time of our algorithm is polynomial. Let us start by defining a notion of *Pareto frontier* as a set of Pareto optimal solutions and introduce Algorithm 1 that finds such set for every delivery instance.

Definition 7. Given a delivery instance $I = \langle [n], G, h \rangle$, its Pareto frontier is a minimal set of allocations \mathcal{F} such that for every PO allocation A there exists $B \in \mathcal{F}$ and permutation of agents π such that $c(A_i) = c(B_{\pi(i)})$, for every $i \in [n]$.

Algorithm. Throughout the algorithm, we keep allocations in the list \mathcal{F} , each allocation sorted in non-increasing cost order. First, we initialize it with just one empty allocation. Then, we look at vertices directly connected to the hub. For each, say u , we run our algorithm on a smaller instance where the graph is just the branch outgoing from h that u is on, and u is the hub. In each allocation in the output, \mathcal{F}' , we add u to the bundle of the first agent. Finally, we combine these allocations, by looking at all possible sorted combinations of allocations in both lists, and keeping only the ones that are not weakly Pareto dominated by any other (where an allocation weakly Pareto dominates another allocation if it Pareto dominates it, or all agents have equal costs in both allocations).

Example 3. We run our algorithm on the instance with 2 agents from Figure 1. First, we run it on two smaller graphs: the vertex a as one and one on the branch containing vertices b, c, d, e, f and g . Vertex a is a leaf, so we get one allocation, i.e., $\mathcal{F} = \{(\{a\}, \emptyset)\}$. When b is the hub, we obtain two allocations: either one agent services g along with all its ancestors and the other agent services c , or one agent services everything. Thus, $\mathcal{F}' = \{(\{b, d, e, f, g\}, \{c\}), (\{b, c, d, e, f, g\}, \emptyset)\}$. Finally, we combine \mathcal{F} with \mathcal{F}' . We consider all four possible combinations. However, one of them, $(\{a, b, d, e, f, g\}, \{c\})$ is Pareto dominated by another, $(\{b, c, d, e, f, g\}, \{a\})$ (the cost of the first agent is the same, but for the second agent it decreases by 1). In conclusion, we return three allocations: $(\{b, d, e, f, g\}, \{a, c\})$, $(\{b, c, d, e, f, g\}, \{a\})$, and $(\{a, b, c, d, e, f, g\}, \emptyset)$. We note that the first one is in fact MMS and PO allocation.

Now, let us prove the correctness of our algorithm.

Theorem 4. Given a delivery instance $I = \langle [n], G, h \rangle$, Algorithm 1 computes its Pareto frontier and runs in time $O((n+2)!m^{3n+2})$, where $m = |E(G)|$.

Proof (sketch). Here, we note two key observations.

For the first, consider an arbitrary instance $\langle [n], G, h \rangle$, and a smaller one obtained by taking a subtree rooted in some vertex $u \in V$, i.e., $\langle [n], T_u, u \rangle$. Then, every PO allocation A in the original instance, must be still PO when we cut it to the smaller instance (i.e., we remove vertices not in T_u). Otherwise, a Pareto improvement on the cut allocation, would also be a Pareto improvement for A . Hence, by looking at all combinations of PO allocations on the branches outgoing from h , we obtain all PO allocations in the original instance.

The second observation is that we do not need to keep two allocations that give the same cost for each agent. The maximum cost of each agent is m . Hence, we will never have more than $(m+1)^n$ different allocations in the list (in fact, since we keep all allocations sorted in non-increasing cost order, this number will be much smaller). Thus, we can combine two frontiers efficiently. \square

We can use Algorithm 1 to obtain the desired allocations.

Theorem 5. There exists an XP algorithm parameterized by n , that given a delivery instance $I = \langle [n], G, h \rangle$, computes an MMS and PO allocation, and decides whether there exist MMS and SO, EF1 and PO, and EF1 and SO allocations.

Proof (sketch). Here, we focus on MMS and PO allocations (see the full version of the paper for the other notions). By Theorem 4, we can obtain a Pareto frontier for every instance. Also, from the proof of Theorem 4, we know that a Pareto frontier contains at most $(m+1)^n$ allocations. Thus, we can search through them to find the leximin optimal one, which will be MMS and PO as well. \square

We note that the proof for EF1 and PO here relies on Cor. 1.

6 Price of Fairness

In this section, we study the *price of fairness*, i.e., the loss to a total cost incurred by requiring EF1 or MMS while delivery tasks located on the vertices of an weighted tree. Specifically, we show tight bounds for the relative amount of such additional cost. The price of fairness has been well studied in the fair division literature and merits exploration in delivery settings as well [Barman *et al.*, 2020; Sun *et al.*, 2023; Bhaskar *et al.*, 2023]. Formally, it is defined as the ratio of the minimum total cost of allocation satisfying given notion to the minimum total cost of any allocation.

Definition 8 (Price of Fairness). Given a fairness concept F and an instance $I = \langle [n], G, h \rangle$, the price of F is given as:

$$\text{PoF}(I) = \frac{\min_{A \in \Pi^n: A \text{ satisfies } F} \sum_{i \in [n]} c(A_i)}{\min_{A \in \Pi^n} \sum_{i \in [n]} c(A_i)}$$

In the following proposition we identify the tight bounds for the price of EF1 and MMS.

Proposition 8. Given a delivery instance $I = \langle [n], G, h \rangle$, it holds that $\text{PoMMS}(I) \leq \frac{n(m-n+1)}{m}$ and $\text{PoEF1}(I) \leq \frac{n(2m-n+1)}{2m}$. Both these bounds are tight.

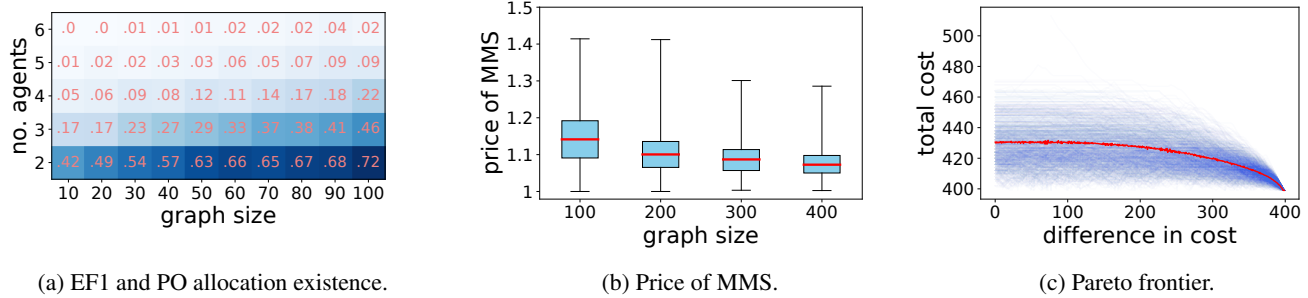


Figure 3: Figure 3a shows the fraction of instances admitting EF1 and PO for different number of agents and sizes of graphs. Figure 3b presents the boxplots with the price of MMS for 2 agents and different graph sizes. In Figure 3c we analyze the trade-off between fairness and the total cost of agents in singular Pareto frontiers for 2 agents and graphs with 400 vertices.

We note that the bound for the price of MMS is lower, which matches our intuition that MMS is more compatible with efficiency in our setting. In the full version of the paper, we also show the hardness of finding cost minimizing allocations satisfying MMS or EF1 and prove that for MMS such allocation has to be PO (which is not the case for EF1). This allows us to use Algorithm 1 to calculate the price of MMS and study it in numerical experiments.

7 Experiments

We now present our experimental results concerning the existence of EF1 and PO allocations and investigate the efficiency loss of fair solutions through their *price of fairness*. We also check the running time of our algorithm, which we report in the full version of the paper [Hosseini *et al.*, 2023a]. In each experiment, we generated trees, uniformly at random, based on Prüfer sequences [Prüfer, 1918] using NetworkX Python library [Hagberg *et al.*, 2008].³ For each experiment and a graph size, we sampled 1,000 trees.

Experiment 1. EF1 and PO existence. First, we checked how often there exists an EF1 and PO allocation. To this end, we generated trees of sizes 10, 20, \dots , 100 and for each tree we run Algorithm 1 for each number of agents from 2 to 6. Based on the output, we checked the number of trees that admit an EF1 and PO allocation. As shown in Figure 3a, the probability of finding an EF1 and PO allocation increases steadily when we increase the size of the graph, but drops sharply when we increase the number of agents. Intuitively, on larger graphs we have more flexibility in how we fairly and efficiently split the vertices. However, when there are more agents, it may still be difficult to satisfy fairness for each of them. We repeat a similar experiment for EF1 and SO as well as MMS and SO allocations in the full version of the paper.

Experiment 2. Price of Fairness. Next, we analyze the price of MMS. To this end, we generated trees of sizes 100, 200, 300, 400 and we run Algorithm 1 for two agents (we also consider larger instances in the full version). Figure 3b illustrates that the median price is around 1.15 for the graphs

of size 100 and it steadily decreases for the larger graphs. These results suggest that as the size of the instance grows, the efficiency loss due to MMS becomes negligible in most cases (at least for a small number of agents).

Experiment 3. Pareto Frontiers. In our final experiment, we analyze the trade-off between fairness and the total cost of agents in singular Pareto frontiers. To this end, we focus on a single size of a graph, 400, and two agents. For each one of 1000 trees generated, we look at each allocation in the Pareto frontier and report the total cost of both agents on y-axis and the difference in costs of agents on x-axis. Then, we connect all such points for allocations in one Pareto frontier to form a partially transparent blue line. By superimposition of all 1000 of such blue lines, we obtain a general view on the distribution of Pareto frontiers. With the thick red line we denote the average total cost, for each difference in costs. We see that particular Pareto frontiers can behave very differently, but the general pattern is quite strong: The total cost does not vary much when the difference in costs is between 0 and 250, however it is much steeper for the larger differences. These findings imply that it is usually not effective to focus on partial fairness as the additional total cost that we incur by guarantying complete fairness instead of partial is not that big. We present a similar experiment for different graph sizes and numbers of agents in the full version of the paper.

8 Conclusions and Future Work

We introduced a novel problem of fair distribution of delivery orders on graphs. We provided a comprehensive characterization of the space of instances that admit fair (EF1 or MMS) and efficient (SO or PO) allocations and—despite proving their hardness—developed an XP algorithm parameterized by the number of agents for each combination of fairness-efficiency notions.

Our work paves the way for future research on developing approximation schemes or perhaps algorithms parameterized by graph characteristics (e.g., maximum degree or diameter) in this domain. Another natural direction is extending our results to cycled and weighted graphs. The model can also be generalized to account for heterogeneous cost functions, capacity constraints of agents, or multiple hubs.

³The code for our experiments is available at: <https://doi.org/10.5281/zenodo.11149658>.

Acknowledgments

Hadi Hosseini acknowledges support from NSF grants #2144413 (CAREER), #2052488, and #2107173. Part of this work was done when Shivika Narang was a visitor at Pennsylvania State University.

References

- [Aziz *et al.*, 2022] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1):1–21, 2022.
- [Barman and Krishnamurthy, 2020] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation*, 8(1):1–28, 2020.
- [Barman *et al.*, 2019] Siddharth Barman, Ganesh Ghalme, Shweta Jain, Pooja Kulkarni, and Shivika Narang. Fair division of indivisible goods among strategic agents. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1811–1813, 2019.
- [Barman *et al.*, 2020] Siddharth Barman, Umang Bhaskar, and Nisarg Shah. Optimal bounds on the price of fairness for indivisible goods. In *Proceedings of the 16th International Conference on Web and Internet Economics (WINE)*, pages 356–369. Springer, 2020.
- [Bhaskar *et al.*, 2021] Umang Bhaskar, AR Sricharan, and Rohit Vaish. On approximate envy-freeness for indivisible chores and mixed resources. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- [Bhaskar *et al.*, 2023] Umang Bhaskar, Neeldhara Misra, Aditi Sethia, and Rohit Vaish. The price of equity with binary valuations and few agent types. In *Proceedings of the 16th International Symposium on Algorithmic Game Theory (SAGT)*, pages 271–289. Springer, 2023.
- [Bilò *et al.*, 2022] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S Zwicker. Almost envy-free allocations with connected bundles. *Games and Economic Behavior*, 131:197–221, 2022.
- [Bouveret *et al.*, 2017] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.
- [Budish, 2011] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- [Caragiannis and Narang, 2023] Ioannis Caragiannis and Shivika Narang. Repeatedly matching items to agents fairly and efficiently. In *Proceedings of the 16th International Symposium on Algorithmic Game Theory (SAGT)*, pages 347–364. Springer, 2023.
- [Caragiannis *et al.*, 2019] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3):1–32, 2019.
- [Chen and Liu, 2020] Xingyu Chen and Zijie Liu. The fairness of leximin in allocation of indivisible chores. *arXiv preprint arXiv:2005.04864*, 2020.
- [Cygan *et al.*, 2015] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015.
- [Ebadian *et al.*, 2022] Soroush Ebadian, Dominik Peters, and Nisarg Shah. How to fairly allocate easy and difficult chores. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 372–380, 2022.
- [Esmaeili *et al.*, 2023] Seyed Esmaeili, Sharmila Duppala, Davidson Cheng, Vedant Nanda, Aravind Srinivasan, and John P Dickerson. Rawlsian fairness in online bipartite matching: Two-sided, group, and individual. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 5624–5632, 2023.
- [Freeman *et al.*, 2019] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Equitable allocations of indivisible goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 280–286, 2019.
- [Ghodsí *et al.*, 2018] Mohammad Ghodsí, MohammadTaghi HajiAghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*, pages 539–556, 2018.
- [Goldman, 1971] Alan J Goldman. Optimal center location in simple networks. *Transportation science*, 5(2):212–221, 1971.
- [Gupta *et al.*, 2022] Anjali Gupta, Rahul Yadav, Ashish Nair, Abhijnan Chakraborty, Sayan Ranu, and Amitabha Bagchi. Fairfoody: Bringing in fairness in food delivery. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 11900–11907, 2022.
- [Hagberg *et al.*, 2008] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), 2008.
- [Hosseini and Searns, 2021] Hadi Hosseini and Andrew Searns. Guaranteeing maximin shares: Some agents left behind. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 238–244, 2021.
- [Hosseini *et al.*, 2022] Hadi Hosseini, Andrew Searns, and Erel Segal-Halevi. Ordinal maximin share approximation for goods. *Journal of Artificial Intelligence Research*, 74:353–391, 2022.

- [Hosseini *et al.*, 2023a] Hadi Hosseini, Shivika Narang, and Tomasz Waś. Fair distribution of delivery orders. *arXiv preprint arXiv:2305.00040*, 2023.
- [Hosseini *et al.*, 2023b] Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Fairly dividing mixtures of goods and chores under lexicographic preferences. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 152–160, 2023.
- [Huang and Lu, 2021] Xin Huang and Pinyan Lu. An algorithmic framework for approximating maximin share allocation of chores. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 630–631, 2021.
- [Kleinberg *et al.*, 1999] Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. In *40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 568–578. IEEE, 1999.
- [Lipton *et al.*, 2004] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, New York, NY, USA, pages 125–131. ACM, 2004.
- [Misra and Nayak, 2022] Neeldhara Misra and Debanuj Nayak. On fair division with binary valuations respecting social networks. In *Proceedings of the 8th Conference on Algorithms and Discrete Applied Mathematics (CAL-DAM)*, pages 265–278. Springer, 2022.
- [Misra *et al.*, 2021] Neeldhara Misra, Chinmay Sonar, PR Vaidyanathan, and Rohit Vaish. Equitable division of a path. *arXiv preprint arXiv:2101.09794*, 2021.
- [Nair *et al.*, 2022] Ashish Nair, Rahul Yadav, Anjali Gupta, Abhijnan Chakraborty, Sayan Ranu, and Amitabha Bagchi. Gigs with guarantees: Achieving fair wage for food delivery workers. *arXiv preprint arXiv:2205.03530*, 2022.
- [O’Dwyer and Timonen, 2009] Ciara O’Dwyer and Virpi Timonen. Doomed to extinction? The nature and future of volunteering for meals-on-wheels services. *VOLUNTAS: International Journal of Voluntary and Nonprofit Organizations*, 20(1):35–49, 2009.
- [Pioro, 2007] Michal Pioro. Fair routing and related optimization problems. In *Proceedings of the 15th International Conference on Advanced Computing and Communications (ADCOM)*, pages 229–235. IEEE, 2007.
- [Prüfer, 1918] Heinz Prüfer. Neuer beweis eines satzes über permutationen. *Archiv für Mathematik und Physik*, 27, 1918.
- [Sánchez *et al.*, 2022] Aitor López Sánchez, Marin Lujak, Frederic Semet, and Holger Billhardt. On balancing fairness and efficiency in routing of cooperative vehicle fleets. In *Proceedings of the 12th International Workshop on Agents in Traffic and Transportation (ATT)*, volume 3173, 2022.
- [Skibski, 2023] Oskar Skibski. Closeness centrality via the condorcet principle. *Social Networks*, 74:13–18, 2023.
- [Sun *et al.*, 2023] Ankang Sun, Bo Chen, and Xuan Vinh Doan. Fairness criteria for allocating indivisible chores: connections and efficiencies. *Autonomous Agents and Multi-Agent Systems*, 37(2):39, 2023.
- [Toth and Vigo, 2002] Paolo Toth and Daniele Vigo. An overview of vehicle routing problems. *The vehicle routing problem*, pages 1–26, 2002.
- [Truszczynski and Lonc, 2020] Mirosław Truszczynski and Zbigniew Lonc. Maximin share allocations on cycles. *Journal of Artificial Intelligence Research*, 69:613–655, 2020.