Delivering Fairly in the Gig Economy

Hadi Hosseini[1] and **Šimon Schierreich**[2]
[1]The Pennsylvania State University
[2]Czech Technical University in Prague
hadi@psu.edu, schiesim@fit.cvut.cz

Abstract

Distributing services, goods, and tasks in the gig economy heavily relies upon on-demand workers (aka agents), leading to new challenges varying from logistics optimization to the ethical treatment of gig workers. We focus on fair and efficient distribution of delivery tasks—placed on the vertices of a graph—among a fixed set of agents. We consider the fairness notion of minimax share (MMS), which aims to minimize the maximum (submodular) cost among agents and is particularly appealing in applications without monetary transfers. We propose a novel efficiency notion—namely, non-wastefulness—that is desirable in a wide range of scenarios and, more importantly, does not suffer from computational barriers. Specifically, given a distribution of tasks, we can, in polynomial time, i) verify whether the distribution is non-wasteful and ii) turn it into an equivalent non-wasteful distribution. Moreover, we investigate several fixed-parameter tractable and polynomial-time algorithms and paint a complete picture of the (parameterized) complexity of finding fair and efficient distributions of tasks with respect to both the structure of the topology and natural restrictions of the input. Finally, we highlight how our findings shed light on computational aspects of other well-studied fairness notions, such as envy-freeness and its relaxations.

1 Introduction

Distributing services, goods, and tasks in today's economy increasingly relies upon on-demand gig workers. In particular, many e-commerce platforms and retail stores utilize freelance workers (in addition to their permanent employees) to distribute goods in an efficient manner. Naturally, this so-called 'gig economy' involves many workers (aka agents), leading to new challenges from logistical and ethical perspectives. While the logistical aspect of this problem has been studied from an optimization perspective [Kleinberg *et al.*, 2001; Toth and Vigo, 2002; Pioro, 2007; Pollner *et al.*, 2022; Knight *et al.*, 2024], little attention has been given to the fair treatment of gig workers.

We focus on the distribution of delivery tasks from a warehouse (the *hub*) that are placed on the vertices of a graph and are connected through an edge (a route) between them. The goal is then to distribute these tasks among a fixed set of agents while adhering to given well-defined notions of fairness and economic efficiency.

A substantial subset of these problems either excludes monetary transfers entirely (e.g., charity organizations) or involves only fixed-salary labor arrangements (e.g., postal service workers). Developing fair algorithms for such scenarios has sparked interest in designing algorithms without money [Procaccia and Tennenholtz, 2013; Ashlagi and Shi, 2014; Narasimhan *et al.*, 2016; Balseiro *et al.*, 2019; Padala and Gujar, 2021] and are notably more challenging compared to those that allow payment-based compensations (i.e., monetary transfers) based on specific tasks [Nisan and Ronen, 1999]. Motivated by this, we primarily focus on a fairness notion of *minimax share* (MMS), which aims to guarantee that no agent incurs a (submodular) cost greater than what they would receive under an (almost) equal distribution. While MMS allocations are guaranteed to exist and are compatible with the economic notion of Pareto optimality (PO), computing such allocations has been shown to be computationally intractable [Hosseini *et al.*, 2025].

1.1 Our Contribution

We generalize the model from the setting where the traversal of each edge costs the same to the *weighted* setting, where the cost of traversing edge can differ. This significantly extends the applicability of the model, as it allows us to capture a broader variety of real-life instances.

Non-Wasteful Allocations. We introduce a new efficiency notion called *non-wastefulness*, which is partly inspired by similar notions in the literature on mechanism design for stable matching [Goto *et al.*, 2016; Kamada and Kojima, 2017; Wu and Roth, 2018; Aziz and Klaus, 2019] and auctions [Kawasaki *et al.*, 2020], and even fair division [Bei *et al.*, 2023; Halpern and Shah, 2019]; however, in these works, non-wastefulness requires that all items are allocated to agents with positive utilities from them. Intuitively, in our context, non-wastefulness states that no delivery order can be reassigned to a different agent so that the original agent is strictly better off and the new worker is not worse off. This fundamental efficiency axiom prevents avoidable duplicate journeys—an obvious choice by delivery agents. Moreover, in contrast to Pareto optimality, it can be verified whether a given allocation is non-wasteful in polynomial time. Additionally, in polynomial-time, *any* distribution can be turned into a non-wasteful one where no agent is worse off. Finally, in Section 4, we formally settle the connection between non-wastefulness and the fairness notions of MMS.

Algorithms for MMS and Non-wasteful Allocations. Our main technical contribution is providing a complete complexity landscape of finding MMS and non-wasteful allocations under various natural parameters. In doing so, we paint a clear dichotomy between tractable and intractable cases. Specifically, in Section 5, we show that if the number of junctions or dead-ends of the topology is bounded, then the problem can be solved efficiently in FPT time, even for weighted instances. In Section 6, we turn our attention to the parameterization by the number of orders and the number of agents, both parameters that are expected to be small in practice. While FPT algorithm for the former is possible even for weighted instances, for the latter, a tractable algorithm is not possible already for two agents. Also, we close an open problem of Hosseini *et al.* [2025] by showing that their XP algorithm for the unweighted case and parameterization by the number of agents is essentially optimal.

The Impact of Topology Structure. Section 7 is then devoted to different restrictions of the topology. The most notable result here is the (in)tractability dichotomy based on the k-path vertex cover, where we prove the existence of FPT algorithms for any weighted instance and $k \leq 3$, and intractability for unweighted instances with $k \geq 4$. Along the way, we identify several polynomial-time algorithms for certain graph families, such as caterpillar graphs, and additional hardness results, such as for unweighted topologies, which are in the distance one to the disjoint union of paths.

For the full version containing all the missing details and additional results, see [Hosseini and Schierreich, 2025].

1.2 Related Work

Fair division of indivisible items is one of the most active areas at the intersection of economics and computer science [Bouveret *et al.*, 2016; Amanatidis *et al.*, 2023]. Different fairness notions are studied in this area, with MMS being one of the prominent ones [Amanatidis *et al.*, 2023; Nguyen and Rothe, 2023]. A relevant literature mostly focus on computational aspects [Bouveret and Lemaître, 2016; Heinen *et al.*, 2018; Nguyen and Rothe, 2023] and existence guarantees [Kurokawa *et al.*, 2018], with special focus on approximations of MMS [Barman and Krishnamurthy, 2020; Xiao *et al.*, 2023; Akrami *et al.*, 2023; Chekuri *et al.*, 2024]. Closest to our work are recent papers of Li *et al.* [2023a] and Wang and Li [2024], which also study submodular costs; however, they do not assume a graph encoding the costs.

Several works also explored fair division on graphs [Christodoulou *et al.*, 2023; Bouveret *et al.*, 2019; Bredereck *et al.*, 2022; Eiben *et al.*, 2023; Bilò *et al.*, 2022; Madathil, 2023; Bouveret *et al.*, 2017; Truszczynski and Lonc, 2020; Li *et al.*, 2023b]. The closest model to ours is the one where we have a graph over items, each agent has certain utility for every item, and the goal is not only to find a fair allocation, but each bundle must additionally form a disjoint and connected sub-graphs.

Finally, there are multiple works exploring fairness in different gig economy contexts, including food delivery [Gupta *et al.*, 2022; Nair *et al.*, 2022] and ride-hailing platforms [Esmaeili *et al.*, 2023; Sánchez *et al.*, 2022]. Nevertheless, these papers mostly focus on experiments and neglect the theoretical study, and the models studied therein are very different.

2 Preliminaries

We use $\mathbb{N}$ to denote the set of positive integers. For an integer $i \in \mathbb{N}$, we set $[i] = \{1, 2, \dots, i\}$ and $[i]_0 = [i] \cup \{0\}$. For notations regarding computational complexity theory (classic and parameterized), we follow the monographs of Arora and Barak [2009] and Cygan *et al.* [2015], respectively.

Distribution of Delivery Orders. In *distribution of delivery orders*, we are given a *topology*, which is an edge-weighted tree $G = (V, E, \omega)$ rooted in a vertex $h \in V$, called a *hub*, and a set of agents $N = \{1, \dots, n\}$. The vertices in $V \setminus \{h\}$ are called *orders*. By m, we denote the number of orders in the given instance. The goal is to find an *allocation* $\pi\colon V \setminus \{h\} \to N$. For the sake of simplicity, we denote by π_i the set of orders allocated to an agent i; that is, $\pi_i = \{v \in V \setminus \{h\} \mid \pi(v) = i\}$. Moreover, we say that π_i is agent i's *bundle* and that an order $v \in \pi_i$ is *serviced* by an agent $i \in N$. By Π, we denote the set of all possible allocations. Formally, an instance of our problem is a triple $\mathcal{I} = (N, G, h)$. We say that an instance $\mathcal{I}$ is *unweighted* if the weights of all edges are the same.

The *cost* of servicing an order $v \in V \setminus \{h\}$, denoted $\mathrm{cost}(v)$, is equal to the length of the shortest path between h and v. A cost for servicing a set $S \subseteq V \setminus \{h\}$ is equal to the length of a shortest walk starting in h, visiting all orders of S, and ending in h, divided by two. Observe that such a walk may also visit some orders that are not in S. It is apparent that the cost function is *submodular* and *identical*.

We use L to denote the number of leaves of the topology G. For a vertex $v \in V$, G^v denotes the sub-tree of G rooted in vertex v, and, for a set $S \subseteq V$, we use W_S to denote the set of all shortest paths with one end in h and a second end in some vertex of S. For graph-theoretical notation not defined here, we follow the monograph of Diestel [2017].

Fairness. We consider *minimax share guarantee* (MMS) as a desired fairness notion. This notion can be seen as a generalization of the famous cake-cutting mechanism and requires that the cost of each agent is, at most, the cost of the worst bundle in the most positive allocation. Formally, the notion is defined as follows.

Definition 1. *An MMS-share of an instance $\mathcal{I}$ of fair distribution of delivery items is defined as*

$$\text{MMS-share}(\mathcal{I}) = \min_{\pi \in \Pi} \max_{i \in [n]} \mathrm{cost}(\pi_i).$$

We say that an allocation π is minimax share *(MMS), if for every agent $i \in N$, it holds that $\mathrm{cost}(\pi_i) \leq \text{MMS-share}(\mathcal{I})$.*

Observe that since the cost functions are identical, we define the MMS-share for the whole instance and not separately for each agent.

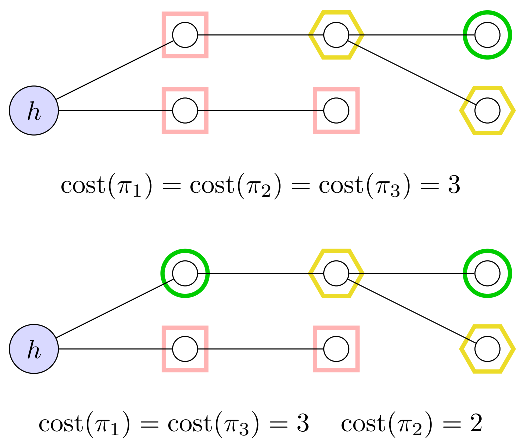

Figure 1: An illustration of non-wastefulness. On the top, we depict an allocation that is not non-wasteful: the red (square) agent services the order of the top branch even though it is not servicing any leaf of this sub-tree. On the bottom, we depict a non-wasteful allocation for the same instance. Observe that in this case, the non-wasteful allocation even strictly improved the cost for the red agent.

Efficiency. We consider several notions of economic efficiency. The most important for us is the Pareto optimality (PO), which, informally, requires there is no other allocation π' such that no agent is worse in π' and at least one agent is strictly better off in π'.

Definition 2. *An allocation π is* Pareto optimal (PO)*, if there is no allocation π' such that for every $i \in N$ $\mathrm{cost}(\pi_i) \geq \mathrm{cost}(\pi'_i)$ and for at least one agent the inequality is strict.*

Sometimes, we also consider utilitarian and egalitarian optimal allocations. In the former, we require that the sum of the costs of all bundles be minimized, while in the latter, we minimize the cost of the most expensive bundle. We defer their formal definitions to the full version of the paper [Hosseini *et al.*, 2025].

3 Non-wasteful Allocations

In this setting, some economic efficiency notions, such as utilitarian optimality, may not be generally compatible with fairness. Moreover, computing an MMS allocation along with Pareto optimality is computationally hard [Hosseini *et al.*, 2025]. Thus, we propose a weaker efficiency notion of non-wastefulness. Informally, a non-wasteful allocation requires that no agent i should be pushed to service an extra order if assigning this order to another agent j reduces the cost of i's bundle without increasing the cost of j's bundle. Formally, we define our efficiency notion as follows; for an illustration of the definition, we refer the reader to Figure 1.

Definition 3. *An allocation π is* non-wasteful *if there is no pair of distinct agents $i, j \in N$ and an order $v \in \pi_i$ such that $\mathrm{cost}(\pi_i \setminus \{v\}) < \mathrm{cost}(\pi_i)$ and $\mathrm{cost}(\pi_j \cup \{v\}) \leq \mathrm{cost}(\pi_j)$.*

Non-wastefulness can be equivalently defined using more graph-theoretical terms as follows. The latter definition is more suitable for our algorithmic results.

Definition 4. *An allocation π is* non-wasteful *if for every order $v \in V \setminus \{h\}$ it holds that if an agent $i \in N$ services v, then i also services some leaf $\ell \in \mathrm{leaves}(G^v)$.*

It is straightforward to see that, in general, non-wasteful allocations are guaranteed to exist. In particular, if we take π so that it allocates all orders to a single agent, then the condition from Definition 4 is satisfied for every internal vertex. Equivalently, in the spirit of the former definition, one can observe that if we reallocate an order from its current bundle to a bundle of some other agent i (recall that such a bundle is empty according to the definition of π), we necessarily increase the cost of i's bundle. Hence, we obtain the following.

Proposition 1. *A non-wasteful allocation is guaranteed to exist and can be found in linear time.*

Our first result shows that we can decide in polynomial time whether a given allocation is non-wasteful or not. This stands in direct contrast with Pareto optimality, which, under the standard theoretical assumptions, cannot admit a polynomial-time algorithm for its associated verification problem [de Keijzer *et al.*, 2009], and makes non-wastefulness arguably one of the fundamental axioms each distribution of delivery orders should satisfy, as agents can check this property basically in hand without the need of extensive computational resources.

The results established in the remainder of this section serve as stepping stones for multiple subsequent sections, where we investigate the algorithmic aspects of non-wastefulness combined with different fairness notions. A naive procedure for verification of non-wastefulness just, for every internal vertex v, checks whether at least one of the leaves in the sub-tree rooted in v is serviced by the agent servicing v.

Theorem 1. *There is an algorithm that, given an instance $\mathcal{I}$ and an allocation π, decides whether π is a non-wasteful allocation in $\mathcal{O}\left(m^2\right)$ time.*

The next important property of non-wastefulness is that, given an allocation π, we can efficiently convert it to a non-wasteful allocation that does not differ from π very significantly and while weakly improving the cost for agents. This result is appealing from the practical perspective, as it can be applied to any existing allocation of delivery tasks with negligible (polynomial) computational overhead. This clearly indicates that non-wastefulness can be very easily used as a layer on top of the current approaches (both algorithmic and manual) for the distribution of delivery tasks without affecting its computability.

Theorem 2. *There is a linear-time algorithm that, given an allocation π, returns a non-wasteful allocation π' such that $\pi_i \cap \mathrm{leaves}(G) = \pi'_i \cap \mathrm{leaves}(G)$ and $\mathrm{cost}(\pi'_i) \leq \mathrm{cost}(\pi_i)$ for every $i \in N$. In other words, in the new non-wasteful allocation π', the set of leaves serviced by an agent $i \in N$ remains the same as in π.*

4 MMS and Non-wasteful Allocations

If we are given an MMS allocation and apply the algorithm from Theorem 2, we obtain a non-wasteful allocation such that the cost of no bundle is increased. Therefore, the new allocation is necessarily both MMS and non-wasteful.

Proposition 2. *Every MMS allocation can be turned into an MMS and non-wasteful allocation in linear time.*

It follows from Proposition 2 that finding MMS and non-wasteful allocations is, from the computational complexity perspective, equivalent to finding an MMS allocation. Therefore, by the result of Hosseini *et al.* [2025], finding MMS and non-wasteful allocation is also computationally intractable, even if the instance is unweighted.

Naturally, the hardness from Hosseini *et al.* [2025] carries over to the more general weighted case, which raises the question of whether there are special topology structures or parameters for which the problem admits tractable algorithms.

In the remainder of this paper, we provide a detailed analysis of the problem's complexity, taking into account both restrictions of the topology and other natural restrictions of the input. Notably, we present the first tractable algorithms for the setting of computing fair and efficient distribution of delivery orders and, in contrast to [Hosseini *et al.*, 2025], some of our positive results also apply to weighted instances, extensively broadening their practical appeal.

Before we dive deep into our results on various topologies, we show several additional auxiliary lemmas that help us simplify the proofs of the following subsections.

First, we show that finding MMS (and non-wasteful) allocation is as hard as deciding whether the MMS-share of an instance is at most a given integer $q \in \mathbb{N}$. This follows from the fact that the cost of the most costly bundle in *all* MMS allocations is the same.

Lemma 1. *Let $\mathfrak{F}$ be a family of instances such that it is* NP*-hard to decide whether the MMS-share of an instance from $\mathfrak{F}$ is at most a given $q \in N$. Then, unless* P = NP*, there is no polynomial time algorithm that finds MMS allocation for all instances from $\mathfrak{F}$.*

The consequence of Lemma 1 is that we can focus only on the complexity of deciding the MMS-share, as the impossibility of a tractable algorithm for finding MMS and non-wasteful allocations follows directly from this lemma and Proposition 2.

Next, we show that one can freely assume that the hub is located on some internal vertex $v \in V(G)$. If this is not the case, then we can move the hub to the single neighbor of the leaf $\ell = h$ and remove ℓ from the instance while preserving the solution of the instance.

Lemma 2. *Let $\mathcal{I} = (N, G = (V, E), h)$ be an instance of fair distribution of delivery orders such that the hub h is a leaf of G and $\mathcal{J}$ be an instance with h removed and with the hub being h's original child $v \in \text{children}(h)$; that is, $\mathcal{J} = (N, (V \setminus \{h\}, E), v)$. Then, it holds that*

$$\text{MMS-share}(\mathcal{I}) = \text{MMS-share}(\mathcal{J}) + \omega(\{h, v\}).$$

Also, by combining the negative result of Hosseini *et al.* [2025, Theorem 1] with Lemma 2, we directly obtain that the intractability of our problem is not caused by a large number of possible routes directly leaving the hub.

Algorithm 1 A dynamic programming algorithm for the computation of an MMS and non-wasteful allocation on instances with a small number of dead-ends.

Input: A problem instance $\mathcal{I} = (G, h, N)$.
Output: MMS-share$(\mathcal{I})$.

return $\min_{Q \subseteq \text{leaves}(G)} \textsc{SolveRec}(n, \text{leaves}(G) \setminus Q, Q)$
function $\textsc{SolveRec}(i, P, Q)$
  if $i = 1$ **and** $\texttt{T}[i, P, Q] = \texttt{undef}$ **then**
    if $P = \emptyset$ **then**
      $\texttt{T}[i, P, Q] \leftarrow \text{cost}(Q)$
    else
      $\texttt{T}[i, P, Q] \leftarrow \infty$
  else if $\texttt{T}[i, P, Q] = \texttt{undef}$ **then**
    if $P \cap Q = \emptyset$ **then**
      $x \leftarrow \min_{P' \subseteq P} \textsc{SolveRec}(i - 1, P \setminus P', P')$
      $\texttt{T}[i, P, Q] \leftarrow \max\{x, \text{cost}(Q)\}$
    else
      $\texttt{T}[i, P, Q] \leftarrow \infty$
  return $\texttt{T}[i, P, Q]$

Corollary 1. *Unless* P = NP*, there is no polynomial-time algorithm that finds an MMS and non-wasteful allocation, even if the instance is unweighted and the degree of the hub is one.*

5 Small Number of Dead-ends or Junctions

We start our algorithmic journey with two efficient algorithms: one for topologies where the number of dead-ends (leaves) is small and one for topologies where the number of junctions (internal vertices) is small. Note that we need to study them separately as none is bounded by another. To see this, assume a star graph with one junction and an arbitrarily large number of dead-ends and, in the opposite direction, a simple path graph with exactly two dead-ends and an arbitrary number of junctions.

We start with an FPT algorithm for the former parameter, that is, the number of leaves L. The algorithm is based on the technique of dynamic programming.

Theorem 3. *When parameterized by the number of leaves L, an MMS and non-wasteful allocation can be found in* FPT *time, even if the instance is weighted.*

Proof Sketch. We prove the result by giving an algorithm running in $2^{\mathcal{O}(L)} \cdot (m + n)^{\mathcal{O}(1)}$ time. The algorithm is based on a dynamic programming approach, and, maybe surprisingly, it does not exploit the topology's structure, as is common for such algorithms, but rather tries to guess for each agent the set of leaves he or she is servicing in an optimal solution. The crucial observation here is that for MMS and non-wastefulness, the agents are interested only in their own bundles. Therefore, we do not need to store the whole partial allocation; rather, we need only the bundle of the currently processed agent and the list of all already allocated orders.

More formally, the core of the algorithm is a dynamic programming table $\texttt{T}[i, P, Q]$, where

- $i \in N$ is the last processed agent,

- $P \subseteq \text{leaves}(G)$ is a subset of leaves allocated to agents $1, \ldots, i-1$, and
- $Q \subseteq \text{leaves}(G) \setminus P$ is a bundle of agent i,

and in each cell of $\mathtt{T}[i, P, Q]$, we store the minimum of the maximum-cost bundle over all partial allocations, where leaves of Q are assigned to agent i, leaves of P are distributed between agents $1, \ldots, i-1$, and leaves of $V \setminus \{h\} \setminus (P \cup Q)$ are unassigned. The computation is then defined as of Algorithm 1. Note that, for the sake of exposition, the code presented computes just the optimal cost. To extend the algorithm so that it also finds an MMS and non-wasteful allocation, we store in each cell a pair (q, π), where q is the minimum cost and π is a partial allocation achieving this cost.

The number of cells of the dynamic programming table is $\mathcal{O}\left(n \cdot 2^L \cdot 2^L\right) \in 2^{\mathcal{O}(L)} \cdot n^{\mathcal{O}(1)}$, and each cell is computed exactly once. The most time-consuming operations of the algorithm are lines 1 and 10, where we, at worst, try all possible subsets of leaves. That is, the overall running time of the algorithm is $2^{\mathcal{O}(L)} \cdot (n+m)^{\mathcal{O}(1)}$ as promised. Note that we made no assumptions about the edge weights. ☐

The structural counterpart of the number of leaves is the number of internal vertices. Again, we show that under this parameterization, our problem is in the complexity class FPT. However, this algorithm is completely different from the previous one and combines an insight into the structure of MMS and non-wasteful solutions with careful guessing and ILP formulation of the carefully designed subproblem.

Theorem 4. *When the instance is parameterized by the number of internal vertices k and the number of different edge weights ψ, an MMS and non-wasteful allocation can be found in FPT time.*

Proof Sketch. Our algorithm combines several ingredients. First, we show a structural lemma that allows us to restrict the number of important agents in terms of the number of internal vertices. Then, for these important agents, we guess their bundles in an optimal solution. Finally, for each guess, we design an integer linear program (ILP) that helps us verify whether our guess is indeed a solution. For the sake of exposition, we show the proof for the unweighted instances; the generalization to instances with a bounded number of different weights is provided in the supplementary material.

Let $\equiv$ be an equivalence relation over the set of leaves such that for a pair $\ell, \ell \in \text{leaves}(G)$ it holds that $\ell \equiv \ell'$ if and only of $\text{parent}(\ell) = \text{parent}(\ell')$. Observe that the relation partitions the leaves into k equivalence classes; we denote them $T_1, \ldots, T_k$. In the following lemma, we show that for each allocation π, there exists an allocation π' where no agent is worse off and which possesses a nice structure.

Lemma 3. *Let π be an allocation. There always exists a* nice *allocation π' such that $\text{cost}(\pi'_i) \leq \text{cost}(\pi_i)$ for every $i \in N$. An allocation is π' is nice if for each pair of distinct agents $i, j \in N$ there exists at most one type $t \in [k]$ so that $|\pi'_i \cap T_t| > 0$ and $|\pi'_j \cap T_t| > 0$.*

The previous lemma implies that there is always an allocation, namely the nice one, where most agents service leaves of exactly one type. To see this, assume that a nice allocation π exists with $\binom{k}{2}+1$ agents servicing at least two different types of leaves. Then, by the Pigeonhole principle, there is necessarily a pair of agents i and j both servicing at least one leaf of some T_t and $T_{t'}$ with $t \neq t'$, which contradicts that π is nice. Consequently, at most $\binom{k}{2}$ agents service leaves of multiple different types, and all other agents services leaves of exactly one type.

In the next phase of the algorithm, we first guess the number $\eta \leq \min\{\binom{k}{2}, n\}$ of important agents, and then for each of agents $i \in [\eta]$, we guess the structure of their bundle. Specifically, for each agent $i \in [\eta]$, the bundle structure is a subset $L_i \subseteq [k]$, where $t \in L_i$ represents that, in a solution π, the agent i services at least one leaf of type t. By Lemma 3, we can assume that all remaining agents $j \in [\eta+1, n]$ are servicing exactly one type of leaves, so we do not need to guess their structure.

To verify whether our guess is correct, we use integer linear programming formulation of the problem. Before introducing the problem's ILP encoding, we guess the MMS-share q of the instance. Note that since the instance is unweighted, there is only a linear number of possible values of q, and we can try all of them in increasing order to obtain the minimum possible q.

In the formulation, we have a non-negative integer variable x_i^t for every $i \in [\eta]$ and every $t \in |T_i|$ representing the number of additional leaves of type t the agent i services. Additionally, we have k variables $y_1, \ldots, y_k$ where each y_j represents the number of agents servicing only the leaves of type T_j. The constraints of the program are as follows (we use $d_t = \text{dist}(\text{parent}(T_t), h)$).

$$\forall i \in [\eta]\colon \qquad \sum_{t \in L_i} (x_i^t + 1 + d_t) \leq q \qquad (1)$$

$$\forall t \in [k]\colon \qquad \sum_{i \in [\eta]\colon\, t \in L_i} (x_i^t + 1) + y_t \cdot (q - d_t) = |T_t| \qquad (2)$$

$$\sum_{t \in [k]} y_t + \eta \leq n \qquad (3)$$

The constraints (1) ensure that the cost of no bundle exceeds the guessed value of the MMS-share. The constraints (2) then secure that all orders are serviced. Finally, due to the constraint (3), the number of agents is correct. Also, observe that we do not use any objective function, as we are only interested in the feasibility of our program. However, we could exploit the objective function to, e.g., find MMS and non-wasteful allocation that minimizes the sum of costs.

For the running time, observe that the number of variables of the program is $\eta \cdot k + k \in \mathcal{O}\left(k^2 \cdot k + k\right) \in \mathcal{O}\left(k^3\right)$. Therefore, the program can be solved in time $k^{\mathcal{O}\left(k^3\right)} \cdot m^{\mathcal{O}(1)}$ by the result of Lenstra Jr. [1983]. There are $2^{\mathcal{O}\left(k^3\right)}$ different guesses we need to verify, and therefore, the overall running time of the algorithm is $2^{\mathcal{O}\left(k^3\right)} \cdot 2^{\mathcal{O}\left(k^3 \log k\right)} \cdot m^{\mathcal{O}(1)} \in 2^{\mathcal{O}\left(k^3 \log k\right)} \cdot m^{\mathcal{O}(1)}$, which is indeed in FPT. ☐

To finalize the complexity picture with respect to the number of internal vertices, in our next result, we show that the parameter the number of different weights cannot be dropped while keeping the problem tractable; in particular, we show that if the number of edge-weights is not bounded, then an efficient algorithm cannot exist already for topologies with a single internal vertex. The reduction is from the 3-PARTITION problem [Garey and Johnson, 1975].

Theorem 5. *Unless* $\mathsf{P} = \mathsf{NP}$*, there is no polynomial-time algorithm that finds an MMS and non-wasteful allocation, even if* G *is a weighted star and the weights are encoded in unary.*

6 Small Number of Agents or Orders

In real-life instances, especially those related to applications such as charity work, it is reasonable to assume that the number of orders or the number of agents is relatively small. Therefore, in this section, we focus on these two parameterizations and provide a complete dichotomy between tractable and intractable cases.

First, assume that our instance possesses a bounded number of orders m. Then, the topology has at most m leaves, and therefore, we can directly use the FPT algorithm from Theorem 3 and efficiently solve even weighted instances.

Corollary 2. *When parameterized by the number of orders* m*, an MMS and non-wasteful allocation can be found in* FPT *time, even if the instance is weighted.*

A more interesting restriction from both the practical and theoretical perspective is when the number of agents is bounded. Our next result rules out the existence of a polynomial-time algorithm already for instances with two agents and uses a very simple topology. The reduction is from a suitable variant of the EQUITABLE PARTITION problem [Deligkas *et al.*, 2024].

Theorem 6. *Unless* $\mathsf{P} = \mathsf{NP}$*, there is no polynomial-time algorithm that finds an MMS and non-wasteful allocation, even if* G *is a weighted star and* $|N| = 2$*.*

For unweighted instances, though, Hosseini *et al.* [2025, Theorem 5] introduced an XP algorithm capable of finding an MMS allocation. That is, if the instance is unweighted, then for every constant number of agents, there is an algorithm that finds an MMS and non-wasteful allocation in polynomial time. Their result raises the question of whether this parameterization admits a fixed-parameter tractable algorithm. We answer this question negatively by showing that, under the standard theoretical assumptions, FPT algorithm is not possible, and therefore, the algorithm of Hosseini *et al.* [2025] is basically optimal. Moreover, the topology created in the following hardness proof is so that if we remove a single vertex, we obtain a disjoint union of paths. This time, we reduce from UNARY BIN PACKING parameterized by the number of bins [Jansen *et al.*, 2013].

Theorem 7. *Unless* $\mathsf{FPT} = \mathsf{W[1]}$*, there is no* FPT *algorithm with respect to the number of agents* $|N|$ *that finds an MMS and non-wasteful allocation, even if the instance is unweighted and the distance to disjoint paths of* G *is one.*

7 Restricted Topologies

In this section, we take a closer look at the computational (in)tractability of fair distribution of delivery orders via different restrictions of the topology. Apart from the theoretical significance of such an approach [Igarashi and Zwicker, 2024; Zhou *et al.*, 2024; Schierreich, 2024], the study is also driven by a practical appeal. It arises in multiple problems involving maps or city topologies that the underlying graph model usually possesses certain structural properties that can be exploited to design efficient algorithms for problems that are computationally intractable in general (see, e.g., [Elkind *et al.*, 2020; Agarwal *et al.*, 2021; Knop and Schierreich, 2023] for a few examples of such studies).

7.1 Star-Like Topologies

Topologies isomorphic to *stars* are particularly interesting for applications where, after processing each order, an agent must return to the hub. One such example is moving companies, where loading a vehicle with more than one order at a time is usually physically impossible.

In contrast to the previous intractability for weighted instances, the following result shows that if G is an unweighted star, then MMS and non-wasteful allocation can be found efficiently.

Proposition 3. *If* G *is a star and the input instance is unweighted, an MMS and non-wasteful allocation can be found in linear time.*

The previous positive results naturally cannot be generalized to the weighted setting as of Theorem 6 already for instances with two agents. However, the hardness in Theorem 6 heavily relies on the fact that the weights of the edges are exponential in the number of orders. This is not a very natural assumption for real-life instances. In practical instances, it is more likely that the weights will be relatively small compared to the number of orders. Fortunately, we show that, for such instances, an efficient algorithm exists for any constant number of agents. The algorithm uses as a subprocedure the MULTI-WAY NUMBER PARTITION problem, where the goal is to partition a set of numbers $\mathcal{A}$ into subsets $A_1, \ldots, A_k$ so that $\max_{i \in [k]} \sum_{a \in A_i} a$ is minimized. This problem is known to admit a pseudo-polynomial time algorithm [Korf, 2009].

Theorem 8. *For every constant* $c \in \mathbb{N}$*, if* G *is a weighted star and* $|N| = c$*, an MMS and non-wasteful allocation can be found in pseudo-polynomial time.*

7.2 Bounded-Depth Topologies

Stars rooted in their center are rather shallow trees; in particular, they are the only family of trees of depth one. It is natural to ask whether the previous algorithms can be generalized to trees of higher depth. In the following result, we show that this is not the case. In fact, our negative result is even stronger and shows that we cannot hope for a tractable algorithm already for unweighted instances of depth two and with diameter four.

Theorem 9. *Unless* $\mathsf{P} = \mathsf{NP}$*, there is no polynomial-time algorithm that finds an MMS and non-wasteful allocation, even if the instance is unweighted, the depth of* G *is two, the diameter of* G *is four, and the* 4*-path vertex cover number of* G *is one.*

The structural parameter 4-path vertex cover mentioned in the previous result can be seen as the minimum number of vertices we need to remove from the topology to obtain a disjoint union of stars. That is, topologies with bounded 4-path vertex cover are generalizations of stars and apply to an even wider variety of real-life instances.

In contrast to the previous hardness result, we show that if the problem is parameterized by the 3-path vertex cover number of the topology, there exists an FPT algorithm. A set of vertices C is called the 3-path vertex cover (3-PVC) if the graph $G' = (V \setminus C, E)$ is a graph of maximum degree one. The size of the smallest possible 3-PVC is then called the 3-path vertex cover number or dissociation number of G [Papadimitriou and Yannakakis, 1982]. This parameter, albeit less common, has been used to obtain tractable algorithms in several areas of artificial intelligence and multiagent systems [Xiao *et al.*, 2017; Grüttemeier *et al.*, 2021; Knop *et al.*, 2022; Grüttemeier and Komusiewicz, 2022], and is also a generalization of the well-known *vertex cover*; if we remove vertex cover vertices, we obtain a graph of maximum degree zero. It is worth mentioning that a minimum size 3-PVC of a tree can be found in polynomial time [Papadimitriou and Yannakakis, 1982]. Therefore, any algorithm for the fair division of delivery orders can first check whether the topology possesses bounded 3-PVC and, if yes, employ our algorithm.

Theorem 10. *If the instance is parameterized by the* 3*-path vertex cover number* ϑ *and the number of different weights* ψ*, combined, an MMS and non-wasteful allocation can be found in* FPT *time.*

The algorithm from Theorem 10 uses as the sub-procedure the FPT algorithm for the parameterization by the number of internal vertices and the number of different edge-weights. In fact, we show that any instance with 3-pvc ϑ and ψ different edge-weights can be transformed to an equivalent instance with $\mathcal{O}(2\vartheta)$ internal vertices and $\mathcal{O}(\psi^2)$ different edge-weights. Such a reduced instance can then be directly solved in FPT time by the algorithm from Theorem 4.

7.3 Topologies with Central Path

When the topology is a simple path, we can find an MMS and non-wasteful allocation in polynomial time: just allocate each leaf to a different agent. Moreover, this approach works even if the instance is weighted.

Proposition 4. *If G is a path, an MMS and non-wasteful allocation can be found in linear time, even if the instance is weighted.*

Therefore, the following set of results explores the complexity picture for instances that are not far from being paths. More specifically, we focus on topologies where all vertices are at a limited distance from a *central path*. Such topologies may appear in practice very naturally, e.g., in instances where the central path is a highway, and the other vertices represent smaller towns along this highway.

Unfortunately, by the intractability results for weighted stars (cf. Theorem 5), we cannot expect any tractable algorithms for topologies with distance to the central path greater or equal to one. Nonetheless, focusing on unweighted instances, we give a polynomial time algorithm for graphs where each vertex is at a distance at most one from the central path; such graphs are commonly known as *caterpillar trees*.

Theorem 11. *If G is a caterpillar tree and the instance is unweighted, an MMS and non-wasteful allocation can be found in polynomial time.*

Proof Sketch. The crucial part of the algorithm is a sub-procedure that, for a given q, returns an allocation π such that the maximum over all bundle costs is q, if such an allocation exists. The sub-procedure is based on the sequential elimination of agents to whom we greedily assign the leaves at the largest distance from the hub h. The algorithm then simply tries all $q \in [m]$ in increasing order and terminates once it reaches q for which the sub-procedure returns a partition. □

The natural subsequent question is whether we can generalize the algorithm from the previous section to larger distances from the central path. It turns out that, without further restriction, this is not the case. In fact, the topology used in the proof of Theorem 9 has all vertices at a distance at most two from the central path, and the created instance is unweighted.

Corollary 3. *Unless* P = NP*, there is no polynomial time algorithm that finds an MMS and non-wasteful allocation, even if all vertices are at a distance at most two from the central path, the central path consists of a single vertex, and the instance is unweighted.*

8 Concluding Remarks

Our work extends the fair delivery problem to settings with weighted edges, proposes non-wastefulness as an efficiency concept, and provides a comprehensive landscape on designing tractable algorithms. We believe that our fixed-parameter and polynomial-time algorithms for computing MMS and non-wasteful allocations may give insights into further strengthening the efficiency notions, e.g., to PO.

Naturally, fair division of delivery orders can extend beyond tree topologies. However, in the presence of cycles, the properties of the model become much more complicated. First, if we allow for arbitrary graphs, already computing the shortest walk needed to service each bundle is computationally intractable, as it requires solving a variant of the travelling salesperson problem [Schierreich and Suchý, 2022; Blažej *et al.*, 2022].

One promising research direction is to study our model in a more dynamic environment. First, one can study the model with *temporal trees*, where some of the edges are not available in every time-step [Schierreich, 2023; Holme and Saramäki, 2019]. A different approach is to study the repeated or temporal distribution of orders, where we are interested in the allocation of (possibly different) orders on each day for a period of days and the goal is to achieve certain fairness guarantees a) for each day and b) for the entire period of time [Cookson *et al.*, 2025; Igarashi *et al.*, 2024].

Acknowledgments

This work was co-funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590). Hadi Hosseini was supported in part by NSF Awards IIS-2144413 and IIS-2107173. Part of the work was done while Šimon Schierreich was visiting Penn State University as a Fulbright-Masaryk Fellow and he acknowledges the support by the Czech Technical University in Prague funded grant No. SGS23/205/OHK3/3T/18.

References

[Agarwal *et al.*, 2021] Aishwarya Agarwal, Edith Elkind, Jiarui Gan, Ayumi Igarashi, Warut Suksompong, and Alexandros A. Voudouris. Schelling games on graphs. *Artificial Intelligence*, 301:103576, 2021.

[Akrami *et al.*, 2023] Hannaneh Akrami, Jugal Garg, Eklavya Sharma, and Setareh Taki. Simplification and improvement of MMS approximation. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence, IJCAI '23*, pages 2485–2493. ijcai.org, 2023.

[Amanatidis *et al.*, 2023] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artificial Intelligence*, 322:103965, 2023.

[Arora and Barak, 2009] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[Ashlagi and Shi, 2014] Itai Ashlagi and Peng Shi. Optimal allocation without money: An engineering approach. In *Proceedings of the 15th ACM Conference on Economics and Computation, EC '14*, pages 351–352. ACM, 2014.

[Aziz and Klaus, 2019] Haris Aziz and Bettina Klaus. Random matching under priorities: Stability and no envy concepts. *Social Choice and Welfare*, 53(2):213–259, 2019.

[Balseiro *et al.*, 2019] Santiago R. Balseiro, Huseyin Gurkan, and Peng Sun. Multiagent mechanism design without money. *Operations Research*, 67(5):1417–1436, 2019.

[Barman and Krishnamurthy, 2020] Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation*, 8(1):5:1–5:28, 2020.

[Bei *et al.*, 2023] Xiaohui Bei, Shengxin Liu, and Xinhang Lu. Fair division with subjective divisibility. In *Proceedings of the 19th Conference on Web and Internet Economics, WINE '23*, page 677, 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.

[Blažej *et al.*, 2022] Václav Blažej, Pratibha Choudhary, Dušan Knop, Šimon Schierreich, Ondřej Suchý, and Tomás Valla. On polynomial kernels for traveling salesperson problem and its generalizations. In *Proceedings of the 30th Annual European Symposium on Algorithms, ESA '22*, volume 244 of *LIPIcs*, pages 22:1–22:16, 2022.

[Bouveret and Lemaître, 2016] Sylvain Bouveret and Michel Lemaître. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290, 2016.

[Bouveret *et al.*, 2016] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In *Handbook of Computational Social Choice*, pages 284–310. Cambridge University Press, 2016.

[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 '17*, pages 135–141. ijcai.org, 2017.

[Bouveret *et al.*, 2019] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. Chore division on a graph. *Autonomous Agents and Multi-Agent Systems*, 33(5):540–563, 2019.

[Bredereck *et al.*, 2022] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. Envy-free allocations respecting social networks. *Artificial Intelligence*, 305:103664, 2022.

[Chekuri *et al.*, 2024] Chandra Chekuri, Pooja Kulkarni, Rucha Kulkarni, and Ruta Mehta. 1/2-approximate MMS allocation for separable piecewise linear concave valuations. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, AAAI '24*, pages 9590–9597. AAAI Press, 2024.

[Christodoulou *et al.*, 2023] George Christodoulou, Amos Fiat, Elias Koutsoupias, and Alkmini Sgouritsa. Fair allocation in graphs. In *Proceedings of the 24th ACM Conference on Economics and Computation, EC '23*, pages 473–488. ACM, 2023.

[Cookson *et al.*, 2025] Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. Temporal fair division. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence, AAAI '25*, pages 13727–13734. AAAI Press, 2025.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, Cham, 2015.

[Deligkas *et al.*, 2024] Argyrios Deligkas, Eduard Eiben, Viktoriia Korchemna, and Šimon Schierreich. The complexity of fair division of indivisible items with externalities. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, AAAI '24*, pages 9653–9661. AAAI Press, 2024.

[Diestel, 2017] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Berlin, Heidelberg, 5th edition, 2017.

[Eiben *et al.*, 2023] Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. Parameterized complexity of envy-free resource allocation in social networks. *Artificial Intelligence*, 315:103826, 2023.

[Elkind *et al.*, 2020] Edith Elkind, Neel Patel, Alan Tsang, and Yair Zick. Keeping your friends close: Land allocation with friends. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI '20*, pages 318–324. ijcai.org, 2020.

[Esmaeili *et al.*, 2023] Seyed A. 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, AAAI '23*, pages 5624–5632. AAAI Press, 2023.

[Garey and Johnson, 1975] Michael R. Garey and David S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397–411, 1975.

[Goto *et al.*, 2016] Masahiro Goto, Atsushi Iwasaki, Yujiro Kawasaki, Ryoji Kurata, Yosuke Yasuda, and Makoto Yokoo. Strategyproof matching with regional minimum and maximum quotas. *Artificial Intelligence*, 235:40–57, 2016.

[Grüttemeier and Komusiewicz, 2022] Niels Grüttemeier and Christian Komusiewicz. Learning Bayesian networks under sparsity constraints: A parameterized complexity analysis. *Journal of Artificial Intelligence Research*, 74:1225–1267, 2022.

[Grüttemeier *et al.*, 2021] Niels Grüttemeier, Christian Komusiewicz, and Nils Morawietz. Efficient Bayesian network structure learning via parameterized local search on topological orderings. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI '21*, pages 12328–12335. AAAI Press, 2021.

[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, AAAI '22*, pages 11900–11907. AAAI Press, 2022.

[Halpern and Shah, 2019] Daniel Halpern and Nisarg Shah. Fair division with subsidy. In *Proceedings of the 12th International Symposium on Algorithmic Game Theory, SAGT '19*, volume 11801 of *LNCS*, pages 374–389. Springer, 2019.

[Heinen *et al.*, 2018] Tobias Heinen, Nhan-Tam Nguyen, Trung Thanh Nguyen, and Jörg Rothe. Approximation and complexity of the optimization and existence problems for maximin share, proportional share, and minimax share allocation of indivisible goods. *Autonomous Agents and Multi-Agent Systems*, 32(6):741–778, 2018.

[Holme and Saramäki, 2019] Petter Holme and Jari Saramäki. *Temporal Network Theory*. Computational Social Sciences. Springer, Cham, 2019.

[Hosseini and Schierreich, 2025] Hadi Hosseini and Šimon Schierreich. The algorithmic landscape of fair and efficient distribution of delivery orders in the gig economy. *CoRR*, abs/2503.16002, 2025.

[Hosseini *et al.*, 2025] Hadi Hosseini, Shivika Narang, and Tomasz Wąs. Fair distribution of delivery orders. *Artificial Intelligence*, 347:104389, 2025.

[Igarashi and Zwicker, 2024] Ayumi Igarashi and William S. Zwicker. Fair division of graphs and of tangled cakes. *Mathematical Programming*, 203(1):931–975, 2024.

[Igarashi *et al.*, 2024] Ayumi Igarashi, Martin Lackner, Oliviero Nardi, and Arianna Novaro. Repeated fair allocation of indivisible items. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, AAAI '24*, pages 9781–9789. AAAI Press, 2024.

[Jansen *et al.*, 2013] Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013.

[Kamada and Kojima, 2017] Yuichiro Kamada and Fuhito Kojima. Stability concepts in matching under distributional constraints. *Journal of Economic Theory*, 168:107–142, 2017.

[Kawasaki *et al.*, 2020] Takehiro Kawasaki, Nathanaël Barrot, Seiji Takanashi, Taiki Todo, and Makoto Yokoo. Strategy-proof and non-wasteful multi-unit auction via social network. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI '20*, pages 2062–2069. AAAI Press, 2020.

[de Keijzer *et al.*, 2009] Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory, ADT '09*, volume 5783 of *LNCS*, pages 98–110. Springer, 2009.

[Kleinberg *et al.*, 2001] Jon M. Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. *Journal of Computer and System Sciences*, 63(1):2–20, 2001.

[Knight *et al.*, 2024] Benjamin Knight, Dmitry Mitrofanov, and Serguei Netessine. The impact of AI technology on the productivity of gig economy workers. In *Proceedings of the 25th ACM Conference on Economics and Computation, EC '24*, page 833. ACM, 2024.

[Knop and Schierreich, 2023] Dušan Knop and Šimon Schierreich. Host community respecting refugee housing. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23*, pages 966–975. ACM, 2023.

[Knop *et al.*, 2022] Dušan Knop, Šimon Schierreich, and Ondřej Suchý. Balancing the spread of two opinions in sparse social networks (student abstract). In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI '22*, pages 12987–12988. AAAI Press, 2022.

[Korf, 2009] Richard E. Korf. Multi-way number partitioning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI '09*, pages 538–543. IJCAI/AAAI, 2009.

[Kurokawa *et al.*, 2018] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM*, 65(2):8:1–8:27, 2018.

[Lenstra Jr., 1983] Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[Li *et al.*, 2023a] Bo Li, Fangxiao Wang, and Yu Zhou. Fair allocation of indivisible chores: Beyond additive costs. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems, NeurIPS '23*, 2023.

[Li *et al.*, 2023b] Lily Li, Evi Micha, Aleksandar Nikolov, and Nisarg Shah. Partitioning friends fairly. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI '23*, pages 5747–5754. AAAI Press, 2023.

[Madathil, 2023] Jayakrishnan Madathil. Fair division of a graph into compact bundles. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence, IJCAI '23*, pages 2835–2843. ijcai.org, 2023.

[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. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI '22*, pages 5122–5128. ijcai.org, 2022.

[Narasimhan *et al.*, 2016] Harikrishna Narasimhan, Shivani Agarwal, and David C. Parkes. Automated mechanism design without money via machine learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI '16*, pages 433–439. IJCAI/AAAI Press, 2016.

[Nguyen and Rothe, 2023] Trung Thanh Nguyen and Jörg Rothe. Complexity results and exact algorithms for fair division of indivisible items: A survey. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence, IJCAI '23*, pages 6732–6740. ijcai.org, 2023.

[Nisan and Ronen, 1999] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing, STOC '99*, pages 129–140. ACM, 1999.

[Padala and Gujar, 2021] Manisha Padala and Sujit Gujar. Mechanism design without money for fair allocations. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI/IAT '21*, pages 382–389, 2021.

[Papadimitriou and Yannakakis, 1982] Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982.

[Pioro, 2007] Michal Pioro. Fair routing and related optimization problems. In *Proceedings of the 15th International Conference on Advanced Computing and Communications, ADCOM '07*, pages 229–235. IEEE, 2007.

[Pollner *et al.*, 2022] Tristan Pollner, Mohammad Roghani, Amin Saberi, and David Wajc. Improved online contention resolution for matchings and applications to the gig economy. In *Proceedings of the 23rd ACM Conference on Economics and Computation, EC '22*, pages 321–322. ACM, 2022.

[Procaccia and Tennenholtz, 2013] Ariel D Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. *ACM Transactions on Economics and Computation*, 1(4):1–26, 2013.

[Sánchez *et al.*, 2022] Aitor López Sánchez, Marin Lujak, Frédéric 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 '22*, volume 3173 of *CEUR Workshop Proceedings*, pages 62–76. CEUR-WS.org, 2022.

[Schierreich and Suchý, 2022] Šimon Schierreich and Ondřej Suchý. Waypoint routing on bounded treewidth graphs. *Information Processing Letters*, 173:106165, 2022.

[Schierreich, 2023] Šimon Schierreich. Maximizing influence spread through a dynamic social network (student abstract). In *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI '23*, pages 16316–16317. AAAI Press, 2023.

[Schierreich, 2024] Šimon Schierreich. Multivariate analysis and structural restrictions in computational social choice. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence, IJCAI '24*, pages 8502–8503. ijcai.org, 2024.

[Toth and Vigo, 2002] Paolo Toth and Daniele Vigo. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3):487–512, 2002.

[Truszczynski and Lonc, 2020] Miroslaw Truszczynski and Zbigniew Lonc. Maximin share allocations on cycles. *Journal of Artificial Intelligence Research*, 69:613–655, 2020.

[Wang and Li, 2024] Fangxiao Wang and Bo Li. Fair surveillance assignment problem. In *Proceedings of the 33rd ACM on Web Conference 2024, WWW '24*, pages 178–186. ACM, 2024.

[Wu and Roth, 2018] Qingyun Wu and Alvin E. Roth. The lattice of envy-free matchings. *Games and Economic Behavior*, 109:201–211, 2018.

[Xiao *et al.*, 2017] Mingyu Xiao, Weibo Lin, Yuanshun Dai, and Yifeng Zeng. A fast algorithm to compute maximum k-plexes in social network analysis. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI '17*, pages 919–925. AAAI Press, 2017.

[Xiao *et al.*, 2023] Mingyu Xiao, Guoliang Qiu, and Sen Huang. MMS allocations of chores with connectivity constraints: New methods and new results. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23*, pages 2886–2888. IFAAMAS, 2023.

[Zhou *et al.*, 2024] Yu Zhou, Tianze Wei, Minming Li, and Bo Li. A complete landscape of EFX allocations on graphs: Goods, chores and mixed manna. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence, IJCAI '24*, pages 3049–3056. ijcai.org, 2024.