

Fair Graphical Resource Allocation with Matching-Induced Utilities

Zheng Chen¹

Bo Li²

Mingming Li³

Guochuan Zhang¹

¹College of Computer Science and Technology, Zhejiang University, Hangzhou, China
{21721122, zgc}@zju.edu.cn

² Department of Computing, The Hong Kong Polytechnic University, Hong Kong
comp-bo.li@polyu.edu.hk

³ Department of Computer Science, City University of Hong Kong, Hong Kong
minming.li@cityu.edu.hk

April 2022

Abstract

We study the fair allocation of graphical resources, where the resources are the vertices in a weighted graph. Each agent's utility equals the weight of the maximum matching in the induced subgraph of the resources allocated to her. We first consider the max-min objective which maximizes the minimum utility over all agents. As the problem is NP-hard, we design a $1/8$ approximation polynomial-time algorithm for any number of agents and then show that the approximation ratio can be improved to $2/3$ when there are two agents. To ensure (approximate) max-min fairness, we observe that significant amount of social welfare is sacrificed inevitably. Therefore, we also study the fairness solution concept of envy-free up to one item (EF1), and show that there is always an EF1 allocation that ensures at least $2/3 + 2/(9n - 3)$ fraction of the maximum possible social welfare, where n is the number of agents. Furthermore, such an allocation can be found in polynomial time.

1 Introduction

Motivated by the real-world deployment of social network-based resource allocation management, we study fair allocations of graphical resources. Fair resource allocation has been actively studied due to its practical applications [Moulin, 2003; Goldman and Procaccia, 2014; Flanigan *et al.*, 2021]. Traditionally, the utilities are assumed to be additive which means an agent's value for a bundle of resources equals the sum of each single item's marginal utility. Some recent works (e.g., [Barman and Krishnamurthy, 2020; Ghodsi *et al.*, 2018]) study subadditive or submodular valuations without considering specific graphical structures of the resources, which happens in the real-world. For example, when a department in some company gets assigned a number of engineers who need to work in pairs, the utility for the department may depend on the way to pair these engineers. Due to the engineers' diverse expertise and personalities, an engineer can make different contribution by pairing with different people. We can use a graph to describe the above problem where the engineers are the vertices and the weight on the edge between two vertices represents the collective contribution by pairing these two engineers. Thus the department's utility for receiving a set of vertices is decided by the maximum matching in the induced subgraph of these vertices. To create a better working environment, we want to fairly allocate the engineers to different departments in the company. There are similar pairwise joint work like long-trip coach driver vs co-driver, and accountant vs cashier.

In this work, we formalize the above scenario as a graphical resource allocation problem, where the resources correspond to the vertices in a graph and each agent's value for the received resources equals

the weight of the maximum matching in the induced subgraph. The graphical nature of resources has been studied in the literature (see, e.g., [Bouveret *et al.*, 2017; Suksompong, 2019; Bilò *et al.*, 2019; Igarashi and Peters, 2019]). However, in this line of research, the graph is used to characterize feasible allocations (such as the resources allocated to each agent should be connected), but the agents still have linear utilities. To ensure fairness, we first consider the objective of finding an allocation so that the minimum utility is maximized, which we call *max-min* objective. One of the most widely studied maximin resource allocation problem is the Santa Claus [Bansal and Sviridenko, 2006], which does not consider the graph structures. Regarding graph structures, our work aligns with the research of balanced graph partition [Miyazawa *et al.*, 2021]. Although there are heuristic algorithms in [Kress *et al.*, 2015; Barketau *et al.*, 2015] to partition a graph where the subgraphs are evaluated by maximum matchings, these algorithms do not have approximation guarantees.

We find that, to achieve the max-min objective, significant amount of social welfare may be inevitably sacrificed. Thereby, we further study the fairness notion of *envy-freeness* (EF) [Foley, 1967]. In an EF allocation, no agent prefers the allocation of another agent to her own. Since the resources are indivisible, such an allocation barely exists. Recent research has focused on achieving its relaxation, *envy-freeness up to one item* (EF1) [Budish, 2011], which requires the envy to be eliminated after removing one item. Our goal is to compute an EF1 allocation that also guarantees high social welfare.

1.1 Our Results

We study the fair allocation of graphical resources when the resources are indivisible and correspond to the vertices, and the agents' valuations are measured by the weights of the maximum matchings in the induced subgraphs. Our main contributions are as follows.

First, we consider the max-min objective in which we aim at partitioning the vertices so that the minimum weight of the maximum matchings in the subgraphs is maximized. As the problem is NP-hard, we design a polynomial-time $1/8$ -approximation algorithm. When the problem only involves two agents, the approximation ratio can be improved to $2/3$. We note that, although max-min objective and its approximations ensure fairness of the agents, it may not be efficient where significant amount of social welfare is inevitably sacrificed.

Thus, we then consider EF1 fairness and study how much social welfare can be ensured by EF1 allocations. For arbitrary number of agents, we show that there is always an EF1 allocation whose social welfare is at least $2/3 + 2/(9n - 3)$ fraction of the maximum social welfare that can be achieved without fairness constraints, where n is the number of agents. Moreover, such an allocation can be found in polynomial time. Note that when $n = 2$, the approximation ratio is $4/5$, and we conjecture that there always exists an EF1 allocation that achieves the maximum social welfare for any number of agents.

Finally, we discuss how to extend our problem to other related settings, such as when the agents have different metrics on the same graph, and when the utilities are measured by other graphical properties.

1.2 Related Works

Two separate research lines are closely related to our work – graph partition and fair division.

Graph Partition. Partitioning graphs into balanced subgraphs has been extensively studied in operations research [Miyazawa *et al.*, 2021] and computer science [Buluç *et al.*, 2016]. There are several popular objectives for evaluating whether a partition is balanced. Among the most prominent ones are the max-min (or min-max) objectives, where the goal is to maximize (or minimize) the total weight of the minimum (or maximum) part. Particularly, the vehicle routing problem (VRP) [Koç *et al.*, 2016], which generalizes the travelling salesperson problem (TSP), is closely related to our work. It asks for an optimal set of routes for a number of vehicles, to visit a set of customers. There are a number of popular variants for the VRP, e.g., the so called heterogeneous vehicle routing problem [Yaman, 2006; Rathinam *et al.*, 2020]. There are many other combinatorial structures studied in graph partitioning problem. For

example, in the min-max tree cover (a.k.a. nurse station location) problem, the task is to use trees to cover an edge-weighted graph such that the largest tree is minimized [Khani and Salavatipour, 2014]. This problem also falls under the umbrella of a more general problem, the graph covering problem, where a set of pairwise disjoint subgraphs (called templates) is used to cover a given graph, such as paths [Farbstein and Levin, 2015], cycles [Traub and Tröbst, 2020], network flows [Borradaile *et al.*, 2017], and matchings [Kress *et al.*, 2015].

Fair Division. Allocating a set of indivisible items among multiple agents is a fundamental problem in the fields of multi-agent systems and computational social choice, and we refer the readers to recent surveys [Amanatidis *et al.*, 2022; Aziz *et al.*, 2022] for more detailed discussion. Envy-freeness (EF) and maximin share fairness (MMS) are two well accepted and extensively studied solution concepts. However, with indivisible items, these requirements are demanding and thus the state-of-the-art research mostly studies their relaxations and approximations. For example, EF1 allocation is studied as a relaxation of EF which always exists [Lipton *et al.*, 2004]. Various constant approximation algorithms for MMS allocations are proposed in [Kurokawa *et al.*, 2018; Garg and Taki, 2021] for additive valuations and in [Barman and Krishnamurthy, 2020; Ghodsi *et al.*, 2018] for subadditive valuations. Our work focuses on indivisible graphical items where agents have combinatorial valuations (neither subadditive nor superadditive) depending on the structural properties. Moreover, all the existing algorithms for non-additive valuations run in polynomial time only if the computation of valuations is assumed to be effortless (i.e., oracles). In contrast, in this work, we aim at designing truly polynomial-time approximation algorithms without valuation oracles.

2 Preliminaries

In our model, a set V of m items is to be allocated to a set N of n agents. The items are connected by a weighted graph $G = (V, E; w)$, where the vertices are the items. Every edge $e \in E$ has a weight $w(e) \geq 0$. Since the valuation only comes from pairs of items, there do not exist any reflexive links (the edge from and pointing to the same item). For any matching (i.e., a set of disjoint edges) $M \subseteq E$, let $w(M) = \sum_{e \in M} w(e)$. For any subgraph G' , let $V(G')$ and $E(G')$ be the sets of vertices and edges in G' respectively. An allocation $X = (X_1, \dots, X_n)$ is a partition of V such that $\cup_{i \in [1, n]} X_i = V$ and $X_i \cap X_j = \emptyset$ for $i \neq j$. If $\cup_{i \in N} X_i \subsetneq V$, the allocation is called *partial*. Each agent i 's utility $u_i(X_i)$ for receiving items X_i equals the weight of the maximum matching in $G[X_i]$, where $G[X_i]$ is the induced subgraph of X_i in G . For ease of exposition, we first study the case when the edge weights are the same to different agents and thus we also use $u(\cdot)$ to denote every agent's utility function. In Section 5, we show how to generalize the model to heterogeneous agents. A problem instance is denoted by $\mathcal{I} = (G, N)$ or $\mathcal{I} = (G, N, w)$ if w is highlighted.

Next we introduce the solution concepts that are studied in this work. To ensure fairness, our first objective is to maximize the minimum value of the agents, i.e., we aim at computing an allocation to achieve

$$\max_{\mathbf{X}=(X_1, \dots, X_n)} \min_{i \in N} u_i(X_i). \quad (1)$$

Given any instance $\mathcal{I}$, we denote the optimal value for (1) as $\text{opt}(\mathcal{I})$. An allocation is α -approximate *max-min fair* if $u_i(X_i) \geq \alpha \cdot \text{opt}(\mathcal{I})$ for every agent $i \in N$. As we will see such an allocation may induce extremely low social welfare, where the social welfare of an allocation $X = (X_1, \dots, X_n)$ is defined as $\text{sw}(X) = \sum_{i \in N} u_i(X_i)$. Note that given any instance $\mathcal{I}$, the best possible social welfare of any allocation is the weight of a maximum matching in the original graph G , which is denoted by $\text{sw}^*(\mathcal{I})$. If the instance $\mathcal{I}$ is clear from the context, we also denote $\text{sw}^*(\mathcal{I})$ and $\text{opt}(\mathcal{I})$ as sw^* and opt for short.

Finally, we introduce EF1 fairness, under which we can guarantee high social welfare.

Definition 2.1 (EF1). An allocation $X = (X_1, \dots, X_n)$ is called *envy-free up to 1 item (EF1)* if for any i and j , there exists $g \in X_j$ such that $u_i(X_i) \geq u_i(X_j \setminus \{g\})$.

3 Max-Min Objective

It can be verified that computing the allocation such that the minimum weight of the maximum matchings in the subgraphs is maximized is an NP-hard problem. To see this, consider a graph with a set of disjoint edges with arbitrary weights. Computing such an allocation is equivalent to solving a partition problem. Thus, we aim at designing polynomial-time approximation algorithms to achieve the max-min objective. Without loss of generality, in this section, we assume $w(e) \geq 1$ for all $e \in E$.

3.1 The Algorithm

Theorem 3.1. For any instance $\mathcal{I} = (G, N)$, Algorithm 1 returns a $1/8$ -approximate max-min fair allocation in polynomial time.

Before proving Theorem 3.1, we first explain the intuition. Given an instance $\mathcal{I} = (G, N)$, to ensure the maximum matching in every subset of vertices to be large, we first try to allocate a maximum matching in the original graph. To be more specific, we first compute a maximum matching in G denoted by $M^* \subseteq E$, and then partition M^* into n bundles $(M_1, \dots, M_n)$ where $w(M_1) \geq \dots \geq w(M_n)$ such that $w(M_n)$ is as large as possible. Since partitioning M^* to maximize the minimum weight $w(M_n)$ is NP-hard, we use the following simple greedy solution, which we call *greedy partition* of M^* .

Greedy Partition. Given a matching M and a number n , partition M into $\Gamma(M) = (M_1, \dots, M_n)$ as follows.

- Sort the edges in M in descending order and rename s.t. $w(e_1) \geq \dots \geq w(e_k)$ where $k = |M|$.
- Initially set $M_1^* = \dots = M_n^* = \emptyset$.
- For $i = 1, \dots, k$, select j such that $w(M_j^*) \leq w(M_{j'}^*)$ for all j' and set $M_j^* = M_j^* \cup \{e_i\}$.
- Reorder $M_1^*, \dots, M_n^*$ from the highest weight to the smallest and denote by $(M_1, \dots, M_n)$ the resulting allocation.

The greedy partition of M^* corresponds to an allocation of vertices where unmatched vertices $V' = V \setminus \cup_{i \in N} V(M_i)$ can be allocated arbitrarily. The good news is that such an allocation achieves the max-min objective when the graph is unweighted.

Lemma 3.2. Letting $(M_1, \dots, M_n)$ be the greedy partition of M^* , $X_i = V(M_i)$ for $i = 1, \dots, n-1$ and $X_n = V(M_n) \cup V'$ form a max-min allocation when the graph is unweighted, i.e., $w(e) = w(e')$ for all $e, e' \in E$.

The bad news is that such an allocation does not have any bounded approximation guarantee when the edges have distinct weights. Consider the following example with two agents and the graph is shown in Figure 1 where $\Delta > 1$ is arbitrarily large. Any allocation with bounded approximation ratio of max-min fairness ensures that every agent has value 1, but by partitioning the maximum matching (which contains a single edge with weight Δ) the smaller bundle has value 0. However, if $|M_1| \geq 2$, such an allocation is $1/2$ -approximate max-min fair.

Lemma 3.3. If $|M_1| \geq 2$, $\Gamma(M^*)$ corresponds to an allocation that is $1/2$ -approximate max-min fair.

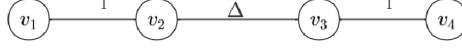


Figure 1: A graph containing four nodes. The weight of edge between node v_2 and v_3 is arbitrarily large.

Now the tricky case is when M_1 contains a single edge, denoted by e^* . The greedy partition fails to approximately maximize the smallest bundle because $w(e^*)$ is too large such that few edges or edges with very small weights can be put in the following M_i 's for $i \geq 2$. One way to overcome the above difficulty is to decrease the weight of e^* and re-compute a maximum matching, through which the advantage of e^* can diminish. Then the problem becomes how much we should decrease the weight of e^* .

For simplicity, we assume all edge weights are powers of 2. This is without much loss of generality which decreases the approximation ratio by at most $1/2$.

Lemma 3.4. *Let $\mathcal{I} = (G, N, w)$ and $\mathcal{I}' = (G, N, w')$ be two instances where $\mathcal{I}'$ is obtained from $\mathcal{I}$ by rounding all edge weights down to the nearest power of 2. If $(X_1, \dots, X_n)$ is an α -approximate allocation in $\mathcal{I}'$, then it is an $\alpha/2$ -approximate allocation in $\mathcal{I}$.*

We prove Lemmas 3.2, 3.3, and 3.4 in Appendix C.

Now we are ready to describe Algorithm 1. We first compute a maximum matching M^* and its greedy partition $\Gamma(M^*) = (M_1, \dots, M_n)$ such that $w(M_1) \geq \dots \geq w(M_n)$. If $|M_1| \geq 2$, then we are safe to output the corresponding partition of vertices so that the approximation ratio is at least $1/4$. If $|M_1| = 1$, we consider two cases. If $w(M_n) \geq 1/2 \cdot w(M_1)$, $w(M_n)$ is still not too small and we can stop the algorithm with a decent approximation ratio. However, if $w(M_n) < 1/2 \cdot w(M_1)$, it means the smallest bundle is far smaller than the largest bundle. Then we update the edge weights: Let H be the edges with weights no smaller than $w(e_1)$ where e_1 is the edge in M_1 , and decrease their weights to $1/2 \cdot w(e_1)$. By repeating the above procedure, eventually we reach an allocation such that $w(M_n) \geq 1/2 \cdot w(M_1)$ or $|M_1| \geq 2$.

Algorithm 1: Max-Min Allocation for n Agents

Input: Instance $\mathcal{I} = (G, N)$ with $G = (V, E; w)$.

Output: Allocation $\mathbf{X} = (X_1, \dots, X_n)$.

1: For all $e \in E$, reset

$$w(e) = 2^{\lfloor \log w(e) \rfloor}.$$

2: Find a maximum matching M^* in G . Denote by V' the set of unmatched vertices.

3: Find the greedy partition $\Gamma(M^*) = (M_1, \dots, M_n)$ of M^* such that $w(M_1) \geq \dots \geq w(M_n)$.

4: **while** $w(M_1) > 2 \cdot w(M_n)$ and G has different weights **do**

5: Let e_1 be the edge in M_1 and $H = \{e \in E \mid w(e) \geq w(e_1)\}$.

6: Let $w(e) = w(e_1)/2$ for all $e \in H$.

7: Re-compute a maximum matching M^* .

8: Re-set V' to be unmatched vertices by M^* .

9: Re-compute the greedy partition $\Gamma(M^*) = (M_1, \dots, M_n)$ such that $w(M_1) \geq \dots \geq w(M_n)$.

10: **end while**

11: Set $X_i = V(M_i)$ for $i = 1, \dots, n-1$.

12: Set $X_n = V(M_n) \cup V'$.

13: Output allocation $(X_1, \dots, X_n)$.

Next, we are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. First, we show Algorithm 1 is well-defined and runs in polynomial time. Every

time when the condition of the **while** loop holds, either the graph has different weights and an allocation is returned or the weights of the heaviest edges are decreased by $1/2^k$ with some $k \geq 1$. Thus the **while** loop is executed $O(\max_{e \in E} \log w(e))$ rounds.

Next we prove the approximation ratio. By Lemma 3.4, we only need to consider the instance where the edge weights are powers of 2 and show the allocation is 1/4-approximate max-min fair. Denote by $O = (O_1, \dots, O_n)$ the optimal solution, where $u(O_1) \geq \dots \geq u(O_n)$ and $\text{opt}(\mathcal{I}) = u(O_n)$. The first time when we reach the **while** loop, if $w(M_1) \leq 2 \cdot w(M_n)$,

$$w(M_n) \geq \frac{1}{2} \cdot w(M_1) \geq \frac{1}{2} \cdot u(O_n) = \frac{1}{2} \cdot \text{opt}(\mathcal{I}),$$

where the second inequality holds because M^* is a maximum matching in G . Thus the allocation is 1/2-approximate max-min fair. If all edges have the same weight, then by Lemma 3.2, the allocation is optimal.

We move into the **while** loop if $w(M_1) > 2 \cdot w(M_n)$ and the edge weights are not identical. Note that $w(M_1) > 2 \cdot w(M_n)$ implies M_1 contains a single edge denoted by e_1 . Otherwise consider the last edge, denoted by e' , added to M_1 in the greedy partition. Then $w(M_1 \setminus \{e'\}) \leq w(M_n)$ and $w(e') \leq w(M_n)$, which implies $w(M_1) \leq 2 \cdot w(M_n)$. After the **while** loop, denote by $\mathcal{I}'$ the instance, by $w'(\cdot)$ the new weights with new utility function $u'(\cdot)$, by $O' = (O'_1, \dots, O'_n)$ the new optimal solution and by M' the maximum matching with greedy partition $(M'_1, \dots, M'_n)$. Then we have the following claim, which is proved in Appendix C.

Claim 3.5. *After each **while** loop, one of the following two cases happens.*

- *Case 1. $w(e_1) \geq 2 \cdot \text{opt}(\mathcal{I})$, then $\text{opt}(\mathcal{I}') = \text{opt}(\mathcal{I})$;*
- *Case 2. $w(e_1) < 2 \cdot \text{opt}(\mathcal{I})$, then $2 \cdot \text{opt}(\mathcal{I}') > \text{opt}(\mathcal{I})$, $w'(M'_1) \leq 2 \cdot w'(M'_n)$.*

By Claim 3.5, the **while** loop will not execute Case 2 or execute Case 2 once following several times of Case 1, and return an allocation. If Case 2 is not executed, then the allocation is 1/2-approximate max-min fair and the analysis is the same with the case when the **while** loop is not executed.

If Case 2 is executed once, then by Claim 3.5,

$$w'(M'_n) \geq \frac{1}{2} \cdot w'(M'_1) \geq \frac{1}{2} \cdot \text{opt}(\mathcal{I}') \geq \frac{1}{4} \cdot \text{opt}(\mathcal{I}).$$

Finally, by Lemma 3.4, the allocation is 1/8-approximate max-min fair for any instance with arbitrary weights and Theorem 3.1 is proved. $\square$

Remark. When $n = 2$, we can improve Algorithm 1 and obtain a better approximation ratio of 2/3. Due to the space limit, we provide the refined algorithm in Appendix B. Furthermore, we complement this refined algorithm with a more efficient implementation that can guarantee 1/2 approximation.

4 Efficient and EF1 Allocations

Recall the example shown in Figure 1. The maximum social welfare is $\text{sw}^* = \Delta$, but any allocation with bounded approximation ratio for max-min fairness has social welfare $2 \ll \Delta$, which means to ensure max-min fairness, we lose significant amount of efficiency. Note that the existence of EF1 allocations is guaranteed by the envy-cycle elimination algorithm designed by Lipton *et al.* [2004]. But the social welfare of the returned allocation does not have any guarantee. In this section, we aim at computing an EF1 allocation that also preserves high social welfare.

Theorem 4.1. *For any instance $\mathcal{I} = (G, N)$, Algorithm 2 returns an EF1 allocation with social welfare at least $(2/3 + 2/(9n - 3)) \cdot \text{sw}^*(\mathcal{I})$ in polynomial time.*

Before designing our algorithm, we first introduce the *EF1-graph*, inspired by the envy-graph introduced in [Lipton *et al.*, 2004]. Given a (partial) allocation $(X_1, \dots, X_n)$, we construct the corresponding EF1-graph $\mathcal{G} = (N, \mathcal{E})$, where the nodes are agents (and thus are used interchangeably) and there is a directed edge from i to j if i envies j (or X_j) for more than one item,

$$u_i(X_i) < u_i(X_j \setminus \{v\}) \text{ for any } v \in X_j.$$

When the agents have identical utility functions, we have the following simple observation.

Observation 4.2. *The EF1-graph $\mathcal{G} = (N, \mathcal{E})$ is acyclic, and the in-degree of the agent with smallest utility is zero.*

Algorithm 2: EF1 Allocation for n Agents

Input: Instance $\mathcal{I} = (G, N)$ with $G = (V, E; w)$.

Output: Allocation $\mathbf{X} = (X_1, \dots, X_n)$.

- 1: Find a maximum matching M^* in G . Denote by V' the set of unmatched vertices by M^* .
 - 2: Find the greedy partition $(M_1, \dots, M_n)$ of edges in M^* such that $w(M_1) \geq \dots \geq w(M_n)$.
 - 3: Set $X_i = V(M_i)$ for $i = 1, \dots, n$.
 - 4: **if** $|M^*| \leq n$ **then**
 - 5: Let $X_n = V(M_n) \cup V'$.
 - 6: Return $(X_1, \dots, X_n)$.
 - 7: **end if**
 - 8: Construct the EF1-graph $\mathcal{G} = (N, \mathcal{E})$ based on $(X_1, \dots, X_n)$.
 - 9: Set Q be the agents with positive in-degree.
 - 10: **for** $i \in Q$ **do**
 - 11: Let $e_i = (v_{i1}, v_{i2})$ be the last edge added to M_i in the greedy-partition procedure.
 - 12: $X_i = X_i \setminus \{v_{i1}\}$ and $V' = V' \cup \{v_{i1}\}$.
 - 13: **end for**
 - 14: **for** $v \in V'$ **do**
 - 15: Let $i = \arg \min_{i \in N} u(X_i)$.
 - 16: Set $X_i = X_i \cup \{v\}$.
 - 17: **end for**
 - 18: Return $(X_1, \dots, X_n)$.
-

Similar with Algorithm 1, in Algorithm 2, we first compute a maximum weighted matching M^* and let the corresponding unmatched vertices be V' . If $|M^*| \leq n$, then by allocating each edge in M^* to a different agent and V' to one agent who has the smallest utility is EF1, since by removing a vertex from an edge, the remaining subgraph does not have edge any more. If $|M^*| > n$, we find a greedy-partition $\Gamma(M^*) = (M_1, \dots, M_n)$ of M^* such that $w(M_1) \geq \dots \geq w(M_n)$. However, by simply assigning $X_i = V(M_i)$ for every i , the allocation may not be EF1. This is illustrated in the tight example after the proof of Theorem 4.1.

To overcome this difficulty, we utilize the EF1-graph $\mathcal{G} = (N, \mathcal{E})$ on the partial allocation $(V(M_1), \dots, V(M_n))$. Let $Q \subseteq N$ be the set of agents who have positive in-degree, i.e., are envied by some agent for more than one item. By Observation 4.2, if $\mathcal{G}$ is nonempty, $Q \neq \emptyset$ and $n \notin Q$. Moreover, since M_n has the smallest weight in the greedy partition $\Gamma(M^*)$, n has an edge to every agent in Q . We first consider the partial allocation after the **for** loop in Step 10, which is denoted by $Y = (Y_1, \dots, Y_n)$.

Lemma 4.3. Y is EF1.

Proof. If $Q = \emptyset$, by definition, the allocation is already EF1. In the following, assume $Q \neq \emptyset$. Note that only the agents in $i \in Q$ has one vertex removed from $V(M_i)$ and for any $i \notin Q$, $Y_i = V(M_i)$. Particularly, $Y_n = V(M_n)$.

Fix any $i \in N \setminus \{n\}$. Let (v_{i1}, v_{i2}) be the edge selected in Step 11, i.e., the edge with smallest weight in M_i . By the definition of greedy partition,

$$u(V(M_n)) \geq u(V(M_i) \setminus \{v_{i1}, v_{i2}\}). \quad (2)$$

We have the following claims, proved in Appendix C.

Claim 4.4. Agent n does not envy agent i for more than one item in the partial allocation Y .

Claim 4.5. Agent i does not envy agent n in Y .

Combining Claims 4.4 and 4.5, we have that for any two agents i and j ,

$$u(Y_i) \geq u(Y_n) \geq u(Y_j \setminus \{v\}) \text{ for some } v \in Y_j,$$

which means i does not envy j for more than one item. This completes the proof of Lemma 4.3. $\square$

To prove the approximation ratio of Algorithm 2, we need the following lemma.

Lemma 4.6. $|M_i| \geq 3$ for all $i \in Q$.

Proof. If the in-degree of agent i is non-zero, then agent n must envy i for more than one item, and

$$u(Y_n) < u(V(M_i) \setminus \{v\}) \text{ for any } v \in V(M_i). \quad (3)$$

First, it is easy to see that $|M_i| \neq 1$ since the removal of any node v makes the remaining utility be 0 and thus Equation (3) does not hold.

Next we show $|M_i| \neq 2$. For the sake of contradiction, assume $M_i = \{e, e'\}$ with $e = (v_1, v_2)$ and $e' = (v'_1, v'_2)$. Without loss of generality, we further assume $w(e) \geq w(e')$. Then it must be that $w(M_n) \geq w(e)$, otherwise e' cannot be added to M_i . Note that since M^* is a maximum weighted matching in G , $\{e, e'\}$ must be a maximum weighted matching in $G[M_i]$. If there exist edges in $G[M_i]$ whose weights are greater than $w(e)$ these edges must be adjacent on the same node, denoted by $\bar{v}$; otherwise they can form another matching with weight greater than $w(M_i)$. Thus by removing $\bar{v}$ from $G[M_i]$, the maximum matching in the remaining graph contains at most one edge, and all the remaining edges have weight at most $w(e)$, which means the maximum matching in $G[V(M_i) \setminus \{\bar{v}\}]$ brings utility no larger than $w(e)$. Therefore,

$$u(V(M_i) \setminus \{\bar{v}\}) \leq w(e) \leq w(M_n),$$

which is a contradiction with Equation (3). Combining the above two cases, we have $|M_i| \geq 3$. $\square$

Now we are ready to prove Theorem 4.1.

Proof of Theorem 4.1. Let $(X_1, \dots, X_n)$ be the allocation returned by Algorithm 2. If the allocation is from Step 6, then it must be EF1. This is because X_n has the smallest value and thus nobody envies n and each of X_i with $1 \leq i \leq n-1$ contains only two nodes which means the removal of one of them brings utility 0 to any agent. It also achieves the optimal social welfare since all edges in M^* are allocated to some edges.

Next we consider the case when the allocation is obtained from Step 18. By Lemma 4.3, after the **for** loop in Step 10, the partial allocation is EF1. To show the final allocation to be EF1, it suffices to

show that the **for** loop in Step 14 preserves EF1. This is true as in each round, only the bundle with smallest value can be allocated one more item whose removal makes it smallest again.

Finally, we consider the social welfare loss. For each agent $i \in Q$, we observe that at most one node will be removed from $V(M_i)$ in the **for** loop in Step 10 and the **for** loop in Step 14 can only increase i 's utility. Since the removed node v_{i1} is from the edge with the smallest weight in M_i , then by Lemma 4.6,

$$u(X_i) \geq \frac{2}{3} \cdot u(V(M_i)) \text{ for all } i \in N \setminus \{n\},$$

Moreover, for agent n and any $i \neq n$,

$$u(X_n) \geq u(M_i \setminus \{(v_{i1}, v_{i2})\}) \geq \frac{2}{3} \cdot u(V(M_i)). \quad (4)$$

Therefore

$$\begin{aligned} \frac{\sum_{i \in N} u(X_i)}{\text{sw}^*} &\geq \frac{\sum_{i \in N \setminus \{n\}} \frac{2}{3} \cdot u(V(M_i)) + u(V(M_n))}{\sum_{i \in N \setminus \{n\}} u(V(M_i))} \\ &= \frac{2}{3} + \frac{1}{3} \cdot \frac{u(V(M_n))}{\sum_{i \in N \setminus \{n\}} u(V(M_i))} \\ &\geq \frac{2}{3} + \frac{1}{3} \cdot \frac{u(V(M_n))}{(\frac{3}{2}(n-1) + 1) \cdot u(V(M_n))} \\ &= \frac{2}{3} + \frac{2}{9n-3}, \end{aligned}$$

where the second inequality is because of Inequality (4) and we complete the proof of Theorem 4.1. $\square$

Tight Example. We show that the analysis in Theorem 4.1 is asymptotically tight. Consider the example in Figure 2, where 2^- means $2 - \epsilon^2$ and 4^+ means $4 + 3(n-1)\epsilon^2$. Let $\epsilon > 0$ be a sufficiently small number, say $1/n^2$. The maximum matching M^* contains all the bold edges and $\text{sw}^* = w(M^*) = 6(n-1) + 4$. By Algorithm 2, the greedy-partition of M^* is $(M_1, \dots, M_n)$ as shown in Figure 2. However, it is not EF1: for $1 \leq i \leq n-1$, by removing any vertex from M_i , the maximum matching in the remaining graph has weight at least $4 + \epsilon > 4 + 3(n-1)\epsilon^2 = w(M_n)$. After the **for** loop in Step 10 in Algorithm 2, for $1 \leq i \leq n-1$, one vertex in each M_i is removed and is reallocated to M_n in the **for** loop in Step 14. Thus the remaining social welfare is at most

$$2 \cdot (2 + \epsilon) \cdot (n-1) + 4 \rightarrow \frac{2}{3} \cdot \text{sw}^*.$$

Remark. By Theorem 4.1, if $n = 2$, the approximation ratio is $4/5$ and when $n \rightarrow \infty$ the approximation ratio is $2/3$. Unfortunately, we were not able to prove an upper bound where the optimal social welfare cannot be achieved by any EF1 allocation. We conjecture that there is always an EF1 allocation that achieves the optimal social welfare sw^* .

5 Extension and Discussion

In the previous sections, we only study the situation when agents have identical valuations. To extend our model to heterogeneous agents, the max-min objective can be generalized to the *maximin share fairness* [Budish, 2011]. Unfortunately, if the agents have different edge weights, the approximation ratio can be arbitrarily small. We recall the related definitions and prove this claim in the Appendix A.1.

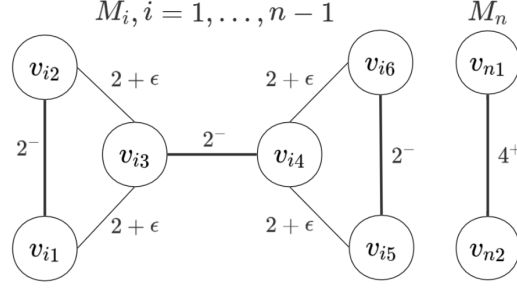


Figure 2: A graph contains n connected components where the first $n - 1$ components are identical as shown by $M_i, i = 1, \dots, n - 1$, and the last component is a single edge as shown by M_n .

Regarding EF1 allocation with heterogeneous agents, the social welfare loss can be as large as $1 - 1/n$ fraction. Moreover, in contrast to the classic setting where the valuations are additive [Caragiannis *et al.*, 2019] or sub-additive [Wu *et al.*, 2021], an allocation that maximizes Nash social welfare does not have any bounded approximation ratio of EF1 fairness. We provide a detailed discussion in Appendix A.

There are many future directions. For example, beyond matchings, we can study other combinatorial properties such as the maximum value of flow or independent set which coincide with real-world applications like image segmentation, financial contagion, and kidney exchange. It is also interesting to consider graphical chores where agents have costs to complete the assigned items.

References

- Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. Fair division of indivisible goods: A survey. *CoRR*, abs/2202.07551, 2022.
- Haris Aziz, Bo Li, Hervé Moulin, and Xiaowei Wu. Algorithmic fair allocation of indivisible items: A survey and new questions. *CoRR*, abs/2202.08713, 2022.
- Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *STOC*, pages 31–40, 2006.
- Maksim Barketau, Erwin Pesch, and Yakov M. Shafransky. Minimizing maximum weight of subsets of a maximum matching in a bipartite graph. *Discret. Appl. Math.*, 196:4–19, 2015.
- Siddharth Barman and Sanath Kumar Krishnamurthy. Approximation algorithms for maximin fair division. *ACM Trans. Economics and Comput.*, 8(1):5:1–5:28, 2020.
- 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. In *ITCS*, volume 124 of *LIPIcs*, pages 14:1–14:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM J. Comput.*, 46(4):1280–1303, 2017.
- Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *IJCAI*, pages 135–141. ijcai.org, 2017.
- E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

- Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In *Algorithm Engineering*, volume 9220 of *Lecture Notes in Computer Science*, pages 117–158, 2016.
- Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Trans. Economics and Comput.*, 7(3):12:1–12:32, 2019.
- Boaz Farbstein and Asaf Levin. Min-max cover of a graph with a small number of parts. *Discret. Optim.*, 16:51–61, 2015.
- Bailey Flanigan, Paul Gözl, Anupam Gupta, Brett Hennig, and Ariel D Procaccia. Fair algorithms for selecting citizens’ assemblies. *Nature*, pages 1–5, 2021.
- D. K. Foley. Resource Allocation and the Public Sector. *Yale Econ. Essays*, 7, 1967.
- Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. *Artificial Intelligence*, 300, 2021.
- Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *EC*, pages 539–556, 2018.
- Jonathan R. Goldman and Ariel D. Procaccia. Spliddit: unleashing fair division algorithms. *SIGecom Exch.*, 13(2):41–46, 2014.
- Ayumi Igarashi and Dominik Peters. Pareto-optimal allocation of indivisible goods with connectivity constraints. In *AAAI*, pages 2045–2052. AAAI Press, 2019.
- M. Reza Khani and Mohammad R. Salavatipour. Improved approximation algorithms for the min-max tree cover and bounded tree cover problems. *Algorithmica*, 69(2):443–460, 2014.
- Çagri Koç, Tolga Bektas, Ola Jabali, and Gilbert Laporte. Thirty years of heterogeneous vehicle routing. *Eur. J. Oper. Res.*, 249(1):1–21, 2016.
- Dominik Kress, Sebastian Meiswinkel, and Erwin Pesch. The partitioning min-max weighted matching problem. *Eur. J. Oper. Res.*, 247(3):745–754, 2015.
- D. Kurokawa, A. Procaccia, and J. Wang. Fair enough: Guaranteeing approximate maximin shares. *Journal of the ACM*, 65(2):8, 2018.
- Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *EC*, pages 125–131. ACM, 2004.
- Flávio Keidi Miyazawa, Phablo F. S. Moura, Matheus J. Ota, and Yoshiko Wakabayashi. Partitioning a graph into balanced connected classes: Formulations, separation and experiments. *Eur. J. Oper. Res.*, 293(3):826–836, 2021.
- Hervé Moulin. *Fair division and collective welfare*. MIT Press, 2003.
- Sivakumar Rathinam, R. Ravi, J. Bae, and Kaarthik Sundar. Primal-dual 2-approximation algorithm for the monotonic multiple depot heterogeneous traveling salesman problem. In *SWAT*, volume 162 of *LIPICs*, pages 33:1–33:13, 2020.
- Warut Suksompong. Fairly allocating contiguous blocks of indivisible items. *Discret. Appl. Math.*, 260:227–236, 2019.

- Vera Traub and Thorben Tröbst. A fast $(2 + 2/7)$ -approximation algorithm for capacitated cycle covering. In *IPCO*, pages 391–404. Springer, 2020.
- Xiaowei Wu, Bo Li, and Jiarui Gan. Budget-feasible maximum nash social welfare allocation is almost envy-free. In *IJCAI*, 2021.
- Hande Yaman. Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Math. Program.*, 106(2):365–390, 2006.

Appendix

The appendix is organized as follows. We first provide a discussion on the extension of our model to heterogeneous agents in Appendix A, and then in Appendix B, we design a new algorithm to compute a $2/3$ -approximation max-min allocation for two agents. Finally, we provide all the missing proofs from the main body in Appendix C.

A Extension and Discussion

In the main body of this work, we only study the situation when the agents are homogeneous. In the following, we discuss the case of heterogeneous agents, where each agent $i \in N$ may have a different weight metric $w_i(\cdot)$ on the edges of graph $G = (V, E)$. Then for a set of vertices $S \subseteq V$, agent i 's utility $u_i(S)$ equals the weight of the maximum matching in $G[S]$ with weights $w_i(\cdot)$.

A.1 Maximin Share Fairness

First, the max-min objective can be extended to the *maximin share fairness* introduced in [Budish, 2011]. Let $\Pi_n(V)$ be the set of all n -partitions of vertices V . An allocation $X = (X_1, \dots, X_n)$ is called α -approximate maximin share fair (α -MMS) if for any i ,

$$u_i(X_i) \geq \alpha \cdot \text{MMS}_i,$$

where

$$\text{MMS}_i = \max_{X \in \Pi_n(V)} \min_{j \in [N]} u_i(X_j).$$

Unfortunately, if the agents have different valuation functions, i.e., different edge weights for the same graph G , the approximation ratio can be arbitrarily small.

Proposition A.1. *Given any $\epsilon > 0$, there exists an instance where in every allocation, there is an agent i who has utility no more than $\epsilon \cdot \text{MMS}_i$.*

Proof. Consider the example as shown in Figure 3. The graph containing four nodes $\{v_1, v_2, v_3, v_4\}$ is allocated to two agents whose valuations (i.e., edge weights) are shown in Figure 3(a) and 3(b) respectively. It can be verified that $\text{MMS}_i = 1$ for both $i = 1, 2$. However, no matter how we allocate the vertices to the agents, one of them receives utility of at most ϵ . $\square$

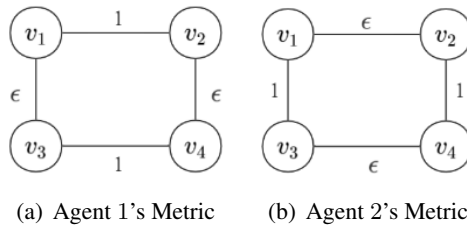


Figure 3: The graph containing four nodes is partitioned and allocated to two agents, who have the opposite valuations.

To overcome the difficulty shown in Proposition A.1, we can consider additive approximation schemes similar with EF1, and also bi-factor approximation algorithm. We leave this as a future research direction.

A.2 EF Fairness and Its Relaxations

We first generalize the definition of EF1 to envy-free up to k items and its approximations.

Definition A.2. For $\alpha \geq 0$, allocation $(X_1, \dots, X_n)$ is α -approximate envy-free up to k items (α -EF k) if for any i and j , there exists $S \subseteq X_j$ such that $|S| \leq k$ and

$$u_i(X_i) \geq \alpha \cdot u_i(X_j \setminus \{g_1, \dots, g_k\}).$$

With heterogeneous agents, it is proved in [Lipton *et al.*, 2004] that an EF1 allocation exists for any monotone and combinatorial valuation. Given the valuation oracles, such an allocation can be found in polynomial time by an envy-free cycle elimination algorithm. Since the matching-induced valuation is monotone and the computation of maximum matching in a graph can be done in polynomial time, an EF1 allocation exists and can be found efficiently.

Similar to Theorem 4.1, we want to find an EF1 allocation which also has high social welfare. Unfortunately, with heterogeneous agents, the fraction of efficiency loss can be as large as $1 - 1/n$. Note that the optimal social welfare sw^* is no longer the maximum matching under a single metric, which can be computed by

$$sw^* = \max_{X \in \Pi_n(V)} \sum_{i \in N} u_i(X_i).$$

Proposition A.3. For any $\epsilon > 0$, there is an instance where every EF1 allocation has social welfare at most $(1/n + \epsilon) \cdot sw^*$.

Proof. If $\epsilon \geq 1 - 1/n$, the proposition holds trivially since no allocation can have social welfare more than sw^* . In the following, we assume $\epsilon < 1 - 1/n$.

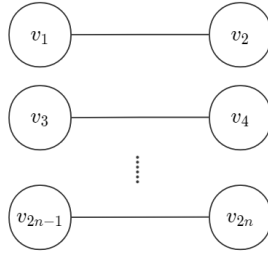


Figure 4: A graph with n disjoint edges is allocated to n agents.

Consider a graph with n disjoint edges, as shown in Figure 4, which is to be allocated to n agents. For each edge, agent 1 has value 1, and the other agents have value ϵ . The maximum social welfare $sw^* = n$ is achieved by allocating all edges to agent 1. However, to guarantee EF1, at most one edge can be given to agent 1. The maximum welfare of EF1 is $1 + (n - 1) \cdot \epsilon$ (each agent receives exactly one edge). The largest ratio is

$$\frac{1 + (n - 1) \cdot \epsilon}{n} < \frac{1}{n} + \epsilon,$$

which completes the proof of the proposition. $\square$

Beyond social welfare, another efficiency notion is Nash social welfare, which is (informally) the product of all agents' utilities. It is proved in [Caragiannis *et al.*, 2019] that under additive valuations, EF1 and Pareto Optimality (PO) are compatible and an allocation that maximizes Nash social welfare is always simultaneously EF1 and PO. Here, an allocation is PO if there is no alternative allocation that makes an agent better off without making anyone worse off. Recently, Wu *et al.* [2021] showed that with subadditive valuations, a Nash social welfare maximizer is still PO but only 1/4-EF1. In our results, we

observe that when the valuations are measured by maximum matchings, Nash social welfare maximizer is PO and EF2 for any matching valuations. However, the Nash social welfare maximizer does not have any bounded approximation guarantee on EF1.

Proposition A.4. *An allocation that maximizes Nash social welfare does not have bounded approximation ratio on EF1.*

Proof. Consider the example shown in Figure 5, where the two agents have different metrics as shown in Figure 5(a) and Figure 5(b) respectively. Let M be a sufficiently large number. The unique allocation that maximizes the Nash social welfare is to assign $X_1 = \{v_3, v_4, v_5, v_6\}$ to agent 1 and $X_2 = \{v_1, v_2\}$ to agent 2. However, for any $v \in X_1$,

$$u_2(X_1 \setminus \{v\}) = M \gg 1 = u_2(X_2).$$

When M goes to infinity, the allocation does not have any bounded approximation ratio. $\square$

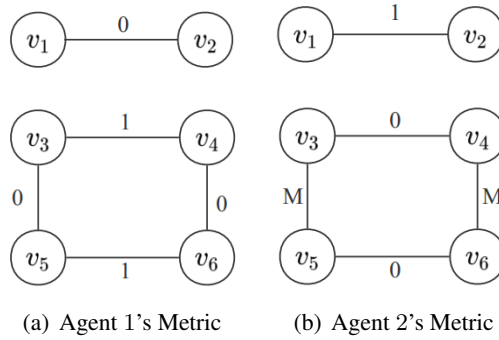


Figure 5: The graph containing a single edge and a square is allocated to two agents.

Note that even with identical valuations, the Nash social welfare maximizing allocation does not guarantee EF1. Consider the example shown in Figure 6, where ϵ is arbitrarily small. To maximize the Nash social welfare, one possible allocation is that $X_1 = \{v_1, v_2\}$ and $X_2 = \{v_3, v_4, v_5, v_6, v_7, v_8\}$. However, this allocation does not guarantee EF1 for agent 1 since the removal of any vertex in X_2 still admits a matching with weight at least 8.5, which is greater than $u_1(X_1) = 8 + \epsilon$.

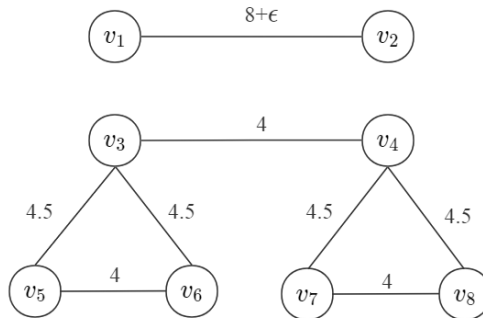


Figure 6: The graph is allocated to two agents with identical valuations. The allocation maximizing Nash social welfare fails to guarantee EF1.

Although Proposition A.4 is disappointing in general, an allocation that maximizes Nash social welfare is PO and EF2, and when the valuations are identical, it ensures 2/3-EF1.

Proposition A.5. *A Nash social welfare maximizing allocation is PO and EF2 for any matching-induced valuations, and is 2/3-EF1 if the valuations are identical.*

Proof. We only prove for the case of two agents, and the proof for arbitrary number of agents is the same. Given any graph $G = (V, E)$, suppose (V_1, V_2) is an allocation that maximizes the Nash social welfare. Denote by $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ the induced subgraphs by V_1 and V_2 respectively. Denote by M_1 and M_2 the maximum matchings in G_1 and G_2 . To show (V_1, V_2) is EF2, we can regard the edges in M_1 and M_2 as items and by the result from [Caragiannis *et al.*, 2019], the allocation must be envy-free up to one edge, which guarantees EF2, since deleting one edge can be achieved by deleting two nodes.

Next we show the allocation is 2/3-EF1 for identical valuations. For the sake of contradiction, suppose that, i.e., for any $v \in V_2$,

$$u(V_1) = w(M_1) < \frac{2}{3} \cdot u(V_2 \setminus \{v\}) \leq \frac{2}{3} \cdot w(M_2).$$

By Lemma 4.6, $|M_2| \geq 3$ and thus the smallest edge in M_2 denoted by e has weight at most $1/3 \cdot w(M_2)$. Thus

$$w(M_1) + w(e) < \frac{2}{3} \cdot w(M_2) + \frac{2}{3} \cdot w(M_2) = w(M_2). \quad (5)$$

Consider a new allocation by assigning edge e to agent 1, then the resulting maximum matchings in each subgraph becomes $M_1 \cup \{e\}$ and $M_2 \setminus \{e\}$. The Nash social welfare of the new allocation is

$$\begin{aligned} & w(M_1 \cup \{e\}) \cdot w(M_2 \setminus \{e\}) \\ &= (w(M_1) + w(e)) \cdot (w(M_2) - w(e)) \\ &= w(M_1) \cdot w(M_2) + (w(M_2) - w(M_1) - w(e)) \cdot w(e) \\ &> w(M_1) \cdot w(M_2), \end{aligned}$$

where the inequality is by (5). However, this is a contradiction with the fact that (V_1, V_2) maximizes the Nash social welfare. $\square$

B Max-Min Allocation for Two Agents

When there are only two agents, Algorithm 1 can be refined as shown in Algorithm 3, and the approximation ratio can be improved to 2/3. Algorithm 3 is similar with Algorithm 1; we first compute a maximum matching M^* and a max-min partition (M_1, M_2) with $w(M_1) \geq w(M_2)$. If $|M_1| \geq 2$, we output the corresponding allocation. Otherwise, in graph G , we directly delete the edge that M_1 contains. We repeat the above procedure until all edges are removed.

Theorem B.1. *Algorithm 3 outputs an allocation that is 2/3-approximate max-min fair in polynomial time.*

Proof. Given an Instance $\mathcal{I} = (G, N, u)$ with $G = (V, E; w)$. Denote by $O = (O_1, O_2)$ the optimal solution, where $u(O_1) \geq u(O_2)$ and $\text{opt}(\mathcal{I}) = u(O_2)$. The first time when we reach the **while** loop, if $w(M_1) \leq 2 \cdot w(M_2)$, allocation $(X_1, X_2) = (V(M_1), V(M_2) \cup V')$ has been output. By Algorithm 3, we have

$$w(M_1) \geq u(O_1) \geq u(O_2) \geq w(M_2).$$

Algorithm 3: Max-Min Allocation for 2 Agents

Input: Instance $\mathcal{I} = (G, N, u)$ with $G = (V, E; w)$.

Output: Allocation $\mathbf{X} = (X_1, X_2)$.

- 1: Find a maximum matching M^* in G . Denote by V' the set of unmatched vertices by M^* .
 - 2: Find the greedy partition (M_1, M_2) of edges in M^* such that $w(M_1) \geq w(M_2)$.
 - 3: Let $Max = w(M_2)$.
 - 4: Set $X_1 = V(M_1)$.
 - 5: Set $X_2 = V(M_2) \cup V'$.
 - 6: **while** $w(M_1) > 2w(M_2)$ **do**
 - 7: By Lemma 3.3. M_1 must contain only one edge. Suppose $M_1 = \{e^*\}$.
 - 8: Delete edge e^* .
 - 9: Re-compute a maximum matching M^* .
 - 10: Re-set V' to be unmatched vertices by M^* .
 - 11: Re-compute the greedy partition (M_1, M_2) of M^* such that $w(M_1) \geq w(M_2)$.
 - 12: **if** $Max < w(M_2)$ **then**
 - 13: $Max = w(M_2)$.
 - 14: Set $X_1 = V(M_1)$.
 - 15: Set $X_2 = V(M_2) \cup V'$.
 - 16: **end if**
 - 17: **end while**
 - 18: Output allocation (X_1, X_2) .
-

Moreover,

$$\begin{aligned} w(M_2) &\geq \frac{1}{3} \cdot (w(M_1) + w(M_2)) \\ &\geq \frac{1}{3} \cdot (u(O_1) + u(O_2)) \\ &\geq \frac{1}{3} \cdot 2 \cdot u(O_2) = \frac{2}{3} \cdot u(O_2). \end{aligned}$$

We move into the **while** loop if $w(M_1) > 2 \cdot w(M_2)$. In such case, M_1 contains only one edge, i.e., $|M_1| = 1$. Suppose $M_1 = \{e^*\}$. There are two subcases:

- Case 1: $e^* \in O_1 \cup O_2$
- Case 2: $e^* \notin O_1 \cup O_2$

First, consider $|M_1| = 1$ and $e^* \in O_1 \cup O_2$. We have: $e^* \in O_1$ and $w(M_2) = u(O_2) = \text{opt}(\mathcal{I})$. The optimal solution has been found and recorded. Therefore, the approximation ratio of the max-min partition is 1. Then the **while** loop is executed for the next round. When $e^* \notin O_1 \cup O_2$, edge e^* is deleted. Let (M'_1, M'_2) be the greedy partition after deleting edge e^* . Then there are two subcases:

- Subcase 1: $|M'_1| \geq 2$
- Subcase 2: $|M'_1| = 1$

For Subcase 1, we will get out of the **while** loop and a $2/3$ -approximate max-min allocation has been determined. For Subcase 2, the **while** loop is executed for the next round. Since we are not sure whether $e^* \in O_1 \cup O_2$, the **while** loop is executed for at most $O(m^2)$ rounds (m is the number of nodes in graph $G(V, E)$). The output (X_1, X_2) is at least $2/3$ -approximate max-min fair allocation. Thus, the theorem holds. $\square$

Lemma B.2. *Algorithm 3 outputs an allocation that is $1/2$ -approximate max-min fair by eliminating at most two edges.*

Proof. Given an Instance $\mathcal{I} = (G, N, u)$ with $G = (V, E; w)$. Denote by $O = (O_1, O_2)$ the optimal solution before eliminating any edge, where $u(O_1) \geq u(O_2)$ and $\text{opt}(\mathcal{I}) = u(O_2)$. Initially, under the maximum matching M' , we find the greedy max-min partition (M_1, M_2) such that $w(M_1) \geq w(M_2)$. If $|M_1| \geq 2$, by Lemma 3.3, (M_1, M_2) is a $1/2$ -approximation max-min partition. Next, consider that M_1 contains only one edge. Suppose $M_1 = \{e_1\}$ and $e_1 \notin O_1 \cup O_2$ (otherwise, by Case 1 in Theorem B.1, $w(M_2) = u(O_2) = \text{opt}(\mathcal{I})$. The optimal solution has been found). If we eliminate edge e_1 , under the re-computed maximum matching, we find the greedy max-min partition (M'_1, M'_2) . Let $O' = (O'_1, O'_2)$ denote the optimal solution after eliminating edge e_1 , where $u(O'_1) \geq u(O'_2)$ and $\text{opt}'(\mathcal{I}') = u(O'_2)$. $M'_1 = \{e'_1\}$ and $e'_1 \notin O'_1 \cup O'_2$. We first show that the two edge e_1 and e'_1 have one common endpoint. By Algorithm 3,

$$w(e_1) > u(O_1) \geq u(O_2) > w(M_2),$$

and

$$w(e'_1) > u(O'_1) \geq u(O'_2) > w(M'_2).$$

Hence, $w(e'_1) > w(M_2)$. Furthermore,

$$w(e_1) + w(e'_1) > w(e_1) + w(M_2).$$

Therefore, edges e_1 and e'_1 make a maximum matching, which implies that there exists an allocation to improve the max-min value from $u(O_2)$ to $w(e'_1)$. It results in a contradiction. Hence, the two edges e_1 and e'_1 have one common endpoint. Furthermore, $e_1 \notin O_1 \cup O_2$ (otherwise, suppose $e_1 \in O_1$. Then O_1 can be replaced by edge e_1 to get larger welfare, which makes a contradiction). Therefore, there are two cases. First, we consider Case 1 (as shown in Figure 7): suppose a_1 and a_2 are two edges in O_2 , and $w(a_1) \geq w(a_2)$. Denote $O_2^* = O_2 / \{a_1, a_2\}$. Thus

$$\max(w(a_1), w(a_2)) \geq \frac{1}{2}(w(a_1) + w(a_2)),$$

and then

$$w(a_1) + u(O_2^*) \geq \frac{1}{2}u(O_2).$$

Accordingly, either

$$w(M_2) \geq w(a_1) + u(O_2^*)$$

or

$$w(M'_2) \geq w(a_1) + u(O_2^*)$$

holds; otherwise, replacing M_2 or M'_2 with $a_1 \cup O_2^*$ can make a matching with larger welfare. Therefore, $w(M_2) \geq 1/2 \cdot u(O_2)$ or $w(M'_2) \geq 1/2 \cdot u(O_2)$. Next, we consider Case 2 (as shown in Figure 8). Suppose after the two edges e_1 and e'_1 have been deleted, under new maximum matching M'' , we find the greedy partition (M''_1, M''_2) . If $|M''_1| \geq 2$, by Theorem B.1, the $2/3$ -approximate max-min partition can be found. If $|M''_1| = 1$, then

$$w(M''_2) \geq w(a_1) + O(B') \geq \frac{1}{2}u(B_2).$$

Thus a $1/2$ -approximate max-min partition has been found, and the lemma holds. $\square$

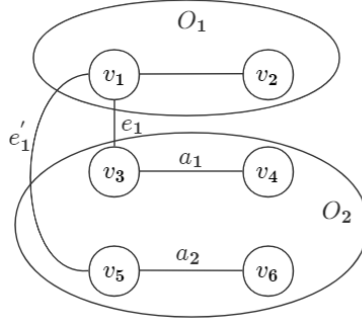


Figure 7: The graph is allocated to two agents with identical valuations. Two large edges e_1 and e'_1 have one common endpoint and their other endpoints connect two distinct edges.

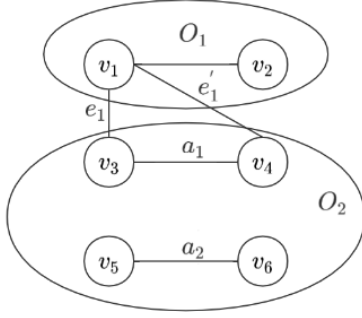


Figure 8: The graph is allocated to two agents with identical valuations. Two large edges e_1 and e'_1 have one common endpoint and their other endpoints are the two endpoints of another same edge.

C Missing Proofs

C.1 Proof of Lemma 3.2

Proof. Without loss of generality, assume all edges have weight 1. In the greedy partition $(M_1, \dots, M_n)$ of M^* , for any $i \in N$,

$$|M_i| \geq |M_n| = \lfloor \frac{|M^*|}{n} \rfloor.$$

Let $(O_1, \dots, O_n)$ be an optimal max-min allocation. If $\text{opt} = |O_n| > |M_n|$, then for all $i \in N$,

$$|O_i| \geq \lfloor \frac{|M^*|}{n} \rfloor + 1.$$

Thus

$$\sum_{i \in N} |O_i| \geq n \cdot \lfloor \frac{|M^*|}{n} \rfloor + n > |M^*|,$$

which is a contradiction with M^* being a maximum matching. $\square$

C.2 Proof of Lemma 3.3

Proof. Denote by $O = (O_1, O_2, \dots, O_n)$ the optimal solution before eliminating any edge, where $u(O_1) \geq u(O_2) \geq \dots \geq u(O_n)$ and $\text{opt}(\mathcal{I}) = u(O_n)$. Under the maximum matching M , consider the greedy partition $(M_1, M_2, \dots, M_n)$, where $u(M_1) \geq u(M_2) \geq \dots \geq u(M_n)$. In greedy partition

procedure, all edges are sorted in descending order of their weights and each time we select the edge with largest weight in the remaining edge set and allocated it to the bundle with the least total utility. If $|M_1| \geq 2$, consider the last edge e added to M_1 , we have $w(M_n) \geq w(e)$, since there exists at least one edge added to M_n before edge e is added to M_1 . Since in the greedy procedure, edges are added to the bundle with less utility, we derive: $w(M_n) \geq w(M_1/e)$. Furthermore, we have

$$\begin{aligned} w(M_n) &\geq \frac{1}{2}(w(e) + w(M_1/e)) \geq \frac{1}{2}w(M_1) \\ &\geq \frac{1}{2n} \sum_{i=1}^n w(M_i) \geq \frac{1}{2n} \sum_{i=1}^n u(O_i) \\ &\geq \frac{1}{2}u(O_n), \end{aligned}$$

and the lemma holds accordingly. $\square$

C.3 Proof of Lemma 3.4

Proof. Let $\mathcal{I}'' = (G, N, w'')$ be the instance obtained from $\mathcal{I}$ by halving all its edge weights. Let opt , opt' and opt'' be the optimal values of instance $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$, respectively. It is easy to see that

$$\text{opt}'' = \frac{1}{2} \cdot \text{opt}.$$

Moreover, the weight of all edges in instance $\mathcal{I}'$ is at least as large as that in instance $\mathcal{I}''$, and thus

$$\text{opt}' \geq \text{opt}'' = \frac{1}{2} \cdot \text{opt}.$$

Finally, since $u_i(X_i) \geq \alpha \cdot \text{opt}'$ for all $i \in N$, then

$$u_i(X_i) \geq \frac{\alpha}{2} \cdot \text{opt},$$

and thus the lemma holds. $\square$

C.4 Proof of Claim 3.5

Proof. We first consider Case 1. For any O_i , if $w(e) < w(e_1)$ for all $e \in E(O_i)$, then $u(O_i)$ does not decrease. If $w(e) \geq w(e_1)$ for some $e \in E(O_i)$, $u(O_i) \geq w(e_1) \geq 2 \cdot \text{opt}(\mathcal{I})$ and after decreasing the weights to $w(e_1)/2$, $u'(O_i) \geq \text{opt}(\mathcal{I})$, implying the existence of an allocation with the minimum utility no smaller than $\text{opt}(\mathcal{I})$, which means $\text{opt}(\mathcal{I}') = \text{opt}(\mathcal{I})$.

Second, we consider Case 2 when $w(e_1) < 2 \cdot \text{opt}(\mathcal{I})$. It is straightforward that $2 \cdot \text{opt}(\mathcal{I}') > \text{opt}(\mathcal{I})$ since $w(e_1) > \text{opt}(\mathcal{I})$ and after decreasing the weights of some edges to $w(e_1)/2$, $u'(O_i) \geq w(e_1)/2 > \text{opt}(\mathcal{I})/2$ (If $e_1 \in O_i$, $u'(O_i) \geq w(e_1)/2$. Otherwise, $u'(O_i) = u(O_i) \geq \text{opt}(\mathcal{I}) > \text{opt}(\mathcal{I})/2$). Next we show $|M'_1| \geq 2$ which implies $w'(M'_1) \leq 2 \cdot w'(M'_n)$. For the sake of contradiction, assume $M'_1 = \{e'_1\}$. Note that at this moment, e'_1 must be an edge with the largest weight in the graph, which means $w'(e'_1) = w'(M'_1) \geq \dots \geq w'(M'_n)$, and thus

$$w'(M') \leq n \cdot w'(e'_1) < n \cdot w'(O'_n) \leq \sum_{i \in N} w'(O'_i).$$

This is a contradiction with M' being a maximum matching in G , which completes the proof. $\square$

C.5 Proofs of Claims 4.4 and 4.5

Proof of Claim 4.4. The claim is straightforward if $i \notin Q$ since there is no edge between n and i . If $i \in Q$, then $Y_i = V(M_i) \setminus \{v_{i1}\}$ and by Inequality (2),

$$u(V(M_n)) \geq u(V(M_i) \setminus \{v_{i1}, v_{i2}\}) = u(Y_i \setminus \{v_{i2}\}),$$

implying n does not envy Y_i for more than one item. □

Proof of Claim 4.5. If $i \notin Q$, both i 's and n 's bundles are not changed in the **for** loop in Step 10, and since M_n has the smallest weight in the greedy partition of M^* ,

$$u(Y_i) = u(V(M_i)) \geq u(V(M_n)) = u(Y_n).$$

If $i \in Q$, since there is an edge from n to i ,

$$u(Y_i) \geq \min_{v \in V(M_i)} u(V(M_i) \setminus \{v\}) > u(V(M_n)),$$

which means i does not envy n . □