ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230346

EFx Budget-Feasible Allocations with High Nash Welfare

Marius Garbea[*], **Vasilis Gkatzelis**[**] **and Xizhi Tan**[***]

Drexel University, Computer Science

Abstract. We study the problem of allocating indivisible items to budget-constrained agents, aiming to provide fairness and efficiency guarantees. Specifically, our goal is to ensure that the resulting allocation is envy-free up to any item (EFx) while minimizing the amount of inefficiency that this needs to introduce. We first show that there exist two-agent problem instances for which no EFx allocation is Pareto-efficient. We, therefore, turn to approximation and use the (Pareto-efficient) maximum Nash welfare allocation as a benchmark. For two-agent instances, we provide a procedure that always returns an EFx allocation while achieving the best possible approximation of the optimal Nash social welfare that EFx allocations can achieve. For the more complicated case of three-agent instances, we provide a procedure that guarantees EFx, while achieving a constant approximation of the optimal Nash social welfare for any number of items.

1 Introduction

One of the central open challenges in the fair division literature is the (approximately) envy-free allocation of indivisible goods [3, 2], i.e., goods that cannot be shared among multiple agents. Envy-freeness is a very natural and well-motivated goal: an allocation of goods among a group of agents is envy-free when no agent would prefer the set of goods allocated to some other agents over the ones allocated to her. However, the fundamental challenge that arises when the goods are indivisible is that envy-freeness may only be achievable by discarding all the goods, which provides no value to any of the agents. An illustrative example of this fact arises when two agents compete over a single indivisible good: if the good is allocated to any of the agents, the other agent is bound to envy her.

To overcome this obstacle, the currently most vibrant line of research in fair division focuses on relaxations of envy-freeness, aiming to provide approximations of this natural fairness property without sacrificing too much value. Two such relaxations that have dominated these efforts are *envy-freeness up to some good* (EF1) (defined explicitly by Budish [7] and implicitly in the earlier work of Lipton et al. [17]) and the more demanding *envy-freeness up to any good* (EFx) (defined by Caragiannis et al. [9]). Both of these notions ensure that any envy from agent i toward agent j would disappear if i were to remove just one of the goods allocated to agent j (her favorite good in the case of EF1 and *any* good in the case of EFx). Therefore, any envy that may exist in EF1 and EFx allocations is up to just one good, which nicely sidesteps the aforementioned single-good illustrative example. In fact, for the well-studied case where the agents' valuations are additive across the goods (i.e., each agent i has a value $v_i(g)$ for each good g and a value $v_i(S) = \sum_{g \in S} v_i(g)$ for any set S of goods), a notable result by Caragiannis et al. [9] shows that returning the highly appealing allocation that maximizes the Nash social welfare (NSW), i.e., the geometric mean of the agents' values, always satisfies EF1. Therefore, unlike envy-freeness, EF1 can be combined with Pareto efficiency. EFx allocations, on the other hand, are much more elusive: if all the goods need to be allocated, then for instances involving more than three agents we do not even know if EFx allocations exist. For instances with up to three agents, Chaudhury et al. [10] proved the existence of EFx allocations using a highly non-trivial procedure that does not guarantee Pareto efficiency.

Rather than assuming that all the goods need to be allocated, a recent line of work has instead proposed procedures that may donate some of the goods, as long as the allocation of the remaining goods (i.e., the ones not donated) satisfies EFx as well as some approximate efficiency guarantees [8, 12, 6]. Notably, using this approach for the case of additive valuations, Caragiannis et al. [8] were able to produce EFx allocations whose Nash social welfare is at least half of the optimal Nash welfare. Therefore, these allocations simultaneously guarantee fairness, in the form of EFx, and efficiency, by approximating a highly desirable Pareto-efficient allocation (unlike other Pareto-efficient allocations, the one maximizing the Nash social welfare is known to provide a natural balance between fairness and efficiency). In this paper, we study the extent to which analogous results that combine fairness and efficiency can be achieved beyond the special case of additive valuations, and we study the more demanding setting where the agents face budget constraints.

The fair allocation of indivisible goods among agents with additive valuations but hard budget constraints was first studied by Wu et al. [19]. In this setting, each good g is associated with a cost $c(g)$, and each agent i has a budget B_i which restricts her to allocations with a total cost within her budget (i.e., she can be allocated a set of goods S only if $\sum_{g \in S} c(g) \leq B_i$). These "budgets" can correspond to actual monetary budgets, but they can also capture several natural space or time limitations beyond that (e.g., when each agent faces, possibly different, capacity constraints for storing the goods allocated to her). Wu et al. [19] showed that maximizing the Nash welfare subject to these budget constraints does not guarantee EF1, in contrast to the setting without budgets [9]. Restricting their attention to the Nash welfare maximizing allocation, they showed that it is approximately EF1 in the sense that agent i's value for j's bundle after removing a good from it can be no more than 4 times her value for her own bundle. Furthermore, they showed that this approximate EF1 bound is tight. Subsequent work on this setting remained focused on the same allocation, proving improved approximate EF1 guarantees for the special case where all agents have identical valuations [15], or approximate-EFx guarantees for the special case where the values

[*] mgarbea@drexel.edu
[**] gkatz@drexel.edu
[***] xizhi@drexel.edu

are all binary (i.e., $v_i(g) \in \{0,1\}$ for all g) [13]. Very recent work by Barman et al. [4] has shown the existence of EF2 allocations (in which no agent envies another agent if he could remove 2 goods from everyone's bundle) even in the budgeted setting.

Our Results. We study the fair allocation of indivisible goods among agents with budget constraints and rather than restricting our attention to the maximum Nash welfare allocation, which does not guarantee EF1 or EFx, we instead guarantee EFx exactly and follow the approach of Caragiannis et al. [8]. That is, we do not assume that all goods need to be allocated (which, in fact, may be infeasible in a setting with budgets) and instead provide efficiency guarantees by proving that our allocations approximate the Nash welfare optimal outcome, which is Pareto-efficient. In the presence of budget constraints, even achieving EF1 without sacrificing too much efficiency becomes a non-trivial problem, which is in contrast to the trivial ways in which EF1 can be achieved in the unrestricted additive valuations case (e.g., using a round-robin procedure). In fact, our first result shows that even for instances involving just three goods and two agents with equal budgets, simultaneously guaranteeing EF1 and Pareto efficiency is infeasible. We then complement this negative result with two positive results for instances involving an arbitrary number of goods and two or three agents of arbitrary budgets.

In Section 3 we focus on instances involving two agents and provide a procedure that returns an EFx allocation with a $\sqrt{0.5} \approx 0.7$ approximation of the optimal Nash welfare. We then show this procedure is optimal in a strong sense: it achieves the best possible approximation that one can guarantee not just for EFx allocations, but even for the more permissive EF1 guarantee. This procedure starts from the budget-feasible allocation with the optimal Nash welfare and, if this is not EFx already, it partitions the bundle of the "envied" agent into two sub-bundles, so that an EFx allocation can be achieved by matching the two agents to two of the three bundles.

In Section 4 we study instances involving three agents. Even if all these agents have the same budget, this is significantly more complicated than unrestricted additive valuations because of the limited ways in which goods can be reallocated across agents' bundles without violating budget feasibility. When each agent's budget is different, this obstacle becomes even more pronounced, as one agent's feasible bundle may be infeasible for another. We provide a procedure that first considers the extreme solution of letting the agents arrive in increasing order of their budgets and choosing their favorite budget-feasible bundle among the remaining goods. We then observe that if in the resulting allocation an agent i envies some agent j who arrived earlier, this allows us to lower i's budget to be equal to agent j's, without sacrificing too much efficiency. Using this intuition, our procedure either reaches an EFx allocation with different budgets or reduces the problem to an instance with equal budgets. Our main result in this section shows that the EFx allocations returned by our procedure always guarantee a constant approximation of the optimal Nash welfare (i.e., an approximation that does not grow with the number of goods or the relative size of the agents' budgets).

Due to space limitations, most of our proofs are deferred to the full version of the paper [16].

Additional related work. Following the approach introduced by Caragiannis et al. [8], i.e., donating goods to limit the envy, recent work has also aimed to achieve EFx while minimizing the *number* of donated goods [12, 6]. This is in contrast to the results of Caragiannis et al. [8] and our results in this paper, which do not optimize for the number of goods donated, as long as the resulting allocation is approximately efficient. Note that minimizing the number of donated goods does not provide any efficiency guarantees in general (e.g., even donating a single good could lead to very low efficiency if that was a highly valued good). Some of this work also ensures that no agent envies the set of donated, which limits the amount of donated value. In fact, even an EFx allocation that donates none of the goods is not necessarily efficient: e.g., if each agent values a distinct subset of goods, an allocation that partitions each such subset "equally" among all the agents is envy-free, yet inefficient. For the special case where all the agents have just two values for the goods and additive values across goods, Amanatidis et al. [1] showed that maximizing the Nash welfare does yield EFx allocations. In general, however, efficiency and fairness cannot be fully achieved simultaneously.

Chaudhury et al. [11] consider the more general class of subadditive valuations and focus on allocations that are approximately, rather than exactly, EFx and that approximate the Nash social welfare optimal outcome. Feldman et al. [14] then capture the tension between EFx and the Nash social welfare by providing tight bounds regarding the trade-off of the approximations achievable for these two notions, both for additive and subadditive valuations.

2 Preliminaries

Given a set of indivisible goods $M = \{1, \ldots, m\}$, we seek to allocate (a subset of) these goods to a group of agents $N = \{1, \ldots, n\}$, where each good $g \in M$ has a cost $c(g) \geq 0$ and each agent $i \in N$ has a budget $B_i \geq 0$. For each subset of goods $S \subseteq M$, we let $c(S) = \sum_{g \in S} c(g)$ represent the total cost of S and say that S is *budget-feasible for agent* i if $c(S) \leq B_i$. Each agent i has a value $v_i(g) \geq 0$ for each good g and her value for being allocated a budget-feasible subset of goods $S \subseteq M$ is additive, i.e., $v_i(S) = \sum_{g \in S} v_i(g)$. Given a set of goods G that may not be budget-feasible for agent i, we use $v_i^{\max}(G) = \max_{S \subseteq G : c(S) \leq B_i} v_i(S)$ to denote the maximum value that the agent can achieve through a subset S of G that is budget-feasible for her. We also use $S_i^{\max}(G)$ to denote the subset of goods in G that is budget-feasible for i and achieves the maximum value, i.e. $v_i^{\max}(G) = v_i(S_i^{\max}(G))$.

An allocation $\mathbf{X} = (\mathrm{X}_1, \ldots, \mathrm{X}_n)$ determines what subset of goods $\mathrm{X}_i \subseteq M$ each agent $i \in N$ gets. Since the goods are indivisible, these sets need to be disjoint, i.e., $\mathrm{X}_i \cap \mathrm{X}_j = \emptyset$ for all $i, j \in N$. We say that an *allocation is budget-feasible* if $c(X_i) \leq B_i$ for all $i \in N$. Note that we do not require an allocation $\mathbf{X}$ to be complete (i.e., that every good is allocated to some agent); in fact, it may be infeasible to do that in a budget-feasible allocation.

Given the agents' budget constraints, an allocation $\mathbf{X}$ is *envy-free* if no agent i would improve her value by replacing their bundle with a subset of another agent's bundle that is budget-feasible for i. Formally, for any two agents $i, j \in N$ we have $v_i(\mathrm{X}_i) \geq v_i^{\max}(\mathrm{X}_j)$, so i has no "justifiable complaints." It is well-known that, even in the absence of any budget constraints, it may be impossible to allocate indivisible goods in an envy-free way, while ensuring that agents receive any positive value.[1] To address this issue, the fair division literature has introduced the following approximate envy-freeness notions, adjusted to our setting:

[1] For example, consider a simple instance with $n = 2$ agents and $m = 2$ goods, such that both agents $i \in \{1, 2\}$ value the first good $v_i(1) = 100$ and the second good $v_i(2) = 50$. If any agent is allocated the first good, the other agent will envy them, so the first good needs to remain unallocated. Given that the first good remains unallocated, though, this similarly implies that whoever receives the second good would be envied, so both goods would have to remain unallocated, leading to a value of 0 for both agents.

Definition 1. *(EF1) An allocation* $\mathbf{X}$ *is* envy-free up to one good *(EF1) with respect to budgets if for any two agents* $i, j \in N$ *and every* $S \subseteq \mathrm{X}_j$ *with* $c(S) \leq B_i$*, there* exists *a good* $g \in S$ *such that*

$$v_i(\mathrm{X}_i) \geq v_i(S \setminus \{g\}).$$

Definition 2. *(EFx) An allocation* $\mathbf{X}$ *is* envy-free up to any good *(EFx) with respect to budgets if for any two agents* $i, j \in N$*, every* $S \subseteq \mathrm{X}_j$ *with* $c(S) \leq B_i$ *and* all *goods* $g \in S$*, we have*

$$v_i(\mathrm{X}_i) \geq v_i(S \setminus \{g\}).$$

We also use the term *EFx-envy* to refer to a violation of the EFx property for some pair of agents, as follows:

Definition 3. *In allocation* $\mathbf{X}$*, an agent* i EFx-envies *another agent* j *if there exists* $S \subseteq \mathrm{X}_j$ *with* $c(S) \leq B_i$ *and* $g \in S$ *such that*

$$v_i(\mathrm{X}_i) < v_i(S \setminus \{g\}).$$

Given these envy notions, we define the EFx-feasibility graph G_{EFx} which indicates the bundles we could assign to each agent such that they would not EFx-envy any other agents.

Definition 4. *Given a subset of agents* $N' \subseteq N$ *and a set* $\mathcal{S}$ *of disjoint bundles of goods (i.e.,* $S_i \cap S_j = \emptyset$ *for all* $S_i, S_j \in \mathcal{S}$*), the EFx-feasibility graph* $G_{EFx}(N', \mathcal{S})$ *is an undirected bipartite graph whose edges are defined as follows:*

$$E(G_{EFx}) = \{(i, S_j) \ : \ v_i^{\max}(S_j) \geq v_i^{\max}(S_k \setminus \{g\}) \ \forall k, \ \forall g \in S_k\}$$

In other words, if there is an edge between some agent i and bundle S_j in G_{EFx}, then allocating S_j to agent i will satisfy the EFx constraint for her, irrespective of how the other bundles are allocated. Therefore, a perfect matching in G_{EFx} corresponds to an EFx allocation (and hence it is "EFx-feasible").

Note that it is always possible to remove any type of envy by keeping all the items unallocated. However, this comes at a great cost in terms of efficiency. To avoid this, we combine envy-freeness guarantees (which capture fairness) with (approximate) guarantees in terms of Pareto efficiency:

Definition 5. *An allocation* $\mathbf{X}$ *is Pareto-efficient if there does not exist any other allocation* $\mathbf{X}'$ *such that* $v_i(\mathrm{X}'_i) > v_i(\mathrm{X}_i)$ *for some agent* i*, and* $v_j(\mathrm{X}'_j) \geq v_j(\mathrm{X}_j)$ *for all agents* $j \in N$*.*

Our first result in Section 3 shows that even for an instance involving just three items and two agents with the same budget, it is impossible to simultaneously guarantee EFx (in fact, not even EF1) and Pareto efficiency. We, therefore, turn to approximation and, among the multiple Pareto-efficient allocations, we use as a benchmark the particularly appealing Pareto-efficient allocation that maximizes the Nash social welfare (NSW) objective:

Definition 6. *The Nash social welfare of an allocation* $\mathbf{X}$ *is*

$$NSW(\mathbf{X}) = \left(\prod_{i \in N} v_i(\mathrm{X}_i) \right)^{1/n}.$$

Specifically, we use the term ρ-efficiency to determine how closely our outcome approximates the optimal Nash social welfare.

Definition 7. *If* $\mathcal{X}$ *is the set of all budget-feasible allocations and* $\mathbf{X}^{OPT} = \operatorname{argmax}_{\mathbf{X} \in \mathcal{X}} NSW(\mathbf{X})$ *is a maximum Nash welfare allocation, then* $\mathbf{X} \in \mathcal{X}$ *is* ρ-efficient*, i.e., a* ρ *approximation, if*

$$NSW(\mathbf{X}) \geq \rho \cdot NSW(\mathbf{X}^{OPT}).$$

Note that, since our benchmark, $\mathbf{X}^{\text{OPT}}$, is Pareto-efficient, any ρ-*efficient* allocation is also approximately Pareto-efficient, with approximation factor ρ. Without loss of generality, we normalize the agents' valuations so that for each agent i, their value in the Nash welfare maximizing solution is equal to 1, i.e., $v_i(\mathbf{X}_i^{\text{OPT}}) = 1$.[2]

Finally, in section 4, we use the term *monopoly value* of some given agent i with respect to a given budget B to refer to the maximum value that this agent could achieve if she were allowed to choose *any* bundle $S \subseteq M$ with cost at most B.

Definition 8. *For each agent* i*, let* $m_i(B)$ *denote the* monopoly value of agent i with respect to budget B*, i.e.*

$$m_i(B) = \max_{S \subseteq M \, : \, c(S) \leq B} v_i(S).$$

3 Instances Involving Two Agents

To exhibit the difficulties that arise in the presence of budgets, we first show that simultaneously guaranteeing EFx (in fact, even EF1) and Pareto efficiency is infeasible. We exhibit this using an instance with just three items and two agents with equal budgets. Furthermore, using the same instance we prove an impossibility result regarding the best approximation of the maximum Nash welfare that is achievable by EF1 allocations (implying the same for EFx allocations as well).

Theorem 1. *There exists a problem instance with two agents of equal budget such that no budget-feasible allocation is both EF1 and Pareto-efficient. Furthermore, for the same instance, no budget-feasible allocation is both EF1 and* $(\sqrt{1/2} + \varepsilon)$*-efficient, for* $\varepsilon > 0$*.*

Proof. Consider the following instance with two agents, where $B_1 = B_2 = 1$, and the set of three items $\{1, 2, 3\}$, whose costs are:

$$c(g) = \begin{cases} 1/2, & \text{for } g \in \{1, 2\} \\ 1, & \text{for } g = 3 \end{cases}$$

The values of the items from each agent's perspective are as follows:

$$v_1(g) = \begin{cases} 1/2, & g \in \{1, 2\} \\ 0, & g = 3 \end{cases} \qquad v_2(g) = \begin{cases} 1 + \varepsilon, & g \in \{1, 2\} \\ 1, & g = 3. \end{cases}$$

For any $\varepsilon < 1$, the budget-feasible allocation that maximizes NSW allocates items $\{1, 2\}$ to agent 1 and item $\{3\}$ to agent 2, leading to a NSW of 1. To verify this, note that if agent 1 were to receive none of these two items her value, and thus also the NSW, would be 0. Also, if agent 1 received just one of these two items, for a value of $1/2$, the value of agent 2 would be at most $1 + \varepsilon$ (since she can afford just one of the remaining two items), leading to NSW less than 1.

However, the NSW maximizing allocation is not EF1 for agent 2, and any EF1 allocation for agent 2 has to leave one item unallocated and is not Pareto-efficient. The budget-feasible EF1 allocation with the largest NSW is $\mathbf{X} = (\{1\}, \{2\})$, with $\text{NSW}(\mathbf{X}) = \sqrt{1/2 \cdot (1 + \varepsilon)} < \sqrt{1/2} + \varepsilon$. Therefore, no budget-feasible EF1 allocation can achieve a NSW approximation of $\sqrt{1/2} + \varepsilon$. □

[2] Note that since the NSW objective is scale-independent, scaling an agent's values for each item by the same constant does not affect the Nash welfare optimal outcome, just its value, so this is without loss of generality.

3.1 EFx Allocations for Two-Agent Instances

We now propose Procedure 1, which takes as input a set of two agents (labeled 1 and 2) and an arbitrary budget-feasible allocation for these two agents, and returns a budget-feasible allocation that is EFx (therefore also EF1) with a NSW at least a $\sqrt{1/2}$ fraction of the original allocation's NSW. Therefore, if we choose the NSW optimal allocation as the original allocation, this procedure returns a $\sqrt{1/2}$-efficient EFx allocation. Note that this is optimal in quite a strong sense, as it achieves the best approximation of the optimal NSW that is possible not just by EFx allocations, but even for the more permissive family of EF1 allocations, as shown in Theorem 1.

Our procedure first checks whether the input allocation is already EFx, in which case it simply returns this allocation, or whether the agents both EFx-envy each other, in which case it "swaps" their bundles (while respecting their budget constraints) and terminates. Otherwise, if just one of them envies the other, for simplicity we reindex the agents so that it is agent 1 who EFx envies agent 2. The procedure then continues with two different approaches based on the "amount" of envy agent 1 has towards agent 2's bundle, X_2.

If agent 1 prefers X_2 at most 2 times more than she likes her own, we proceed as follows: agent 1 repeatedly removes her least valued good g from X_2 and sets it aside in a separate bundle, R. The procedure terminates when it can find a "matching" between the two agents and two of the three bundles (X_1, X_2, or R) that yields an EFx allocation. The crucial observation is that agent 1 removes items from X_2 only while it EFx-envies that bundle (otherwise an EFx allocation is reached), and since she always removes her least valued item from it, even if after some removal she becomes EFx-feasible with another bundle, she will remain EFx-feasible with X_2. This ensures that at some point if no matching has been found already, at least two of the three bundles will be EFx-feasible for agent 1 (specifically X_1 and X_2 as shown in Lemma 1), allowing us to find a matching by giving agent 2 the one she prefers. Note that the existence of this matching relies on the assumption that $v_1(\mathrm{X}_1) \geq \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$.

If this assumption is not true, we use a different approach - the leximin++ procedure (Algorithm 2) of Plaut and Roughgarden [18]. Intuitively, this procedure splits a bundle of items between two agents such that an EFx allocation can be achieved when agents have general valuations (which includes our budget-feasible $v^{\max}$ function). We use this to split X_2 into two parts such that agent 1 would be EFx-feasible with both if these bundles were the only ones participating in the matching. This similarly allows us to prove the existence of at least two edges for agent 1 in the EFx-feasibility graph, which implies the existence of a perfect matching.

In order to prove the desired efficiency guarantees, we show a stronger statement in terms of individual value guarantees. More specifically, we show that Procedure 1 always finds a perfect matching in which agent 1 (the envying agent) gets a weakly higher value than in the input allocation, whereas agent 2 (the envied agent) gets at least half of her original value.

For the following lemmas, let g_t be the item removed in iteration t of the while loop for the case when $v_1(\mathrm{X}_1) \geq \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$. Also, let X_2^t and R^t be the state of the bundles at the end of this iteration (after moving g_t from X_2^t to R^t). Note that if the input allocation $\mathbf{X}$ is already EFx, or if both agents EFx-envy each other in $\mathbf{X}$, we can immediately return a budget-feasible EFx allocation in which both agents (weakly) improve their value. If none of these two statements are true, it must be that one agent (agent 1 by the reindexing in line 4) EFx-envies the other, while the second agent does not EFx-envy the first. For this reason, for the rest of the analysis, we assume that agent 1 EFx-envies agent 2.

Procedure 1: (EFx-2A) EFx allocation for 2 agents

1 **Input**: Set of two agents $\{1, 2\}$; budget-feasible allocation $\mathbf{X}$ for these agents
2 **if** $\mathbf{X}$ *is EFx* **then return** $\mathbf{X}$
3 **if** *both of the agents EFx-envy each other in* $\mathbf{X}$ **then return** $(S_1^{\max}(\mathrm{X}_2), S_2^{\max}(\mathrm{X}_1))$
4 Reindex the agents so that agent 1 EFx-envies agent 2
5 **if** $v_1(\mathrm{X}_1) \geq \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$ **then**
6 $R \leftarrow \emptyset$
7 **while** $G_{EFx}(\{1, 2\}, \{\mathrm{X}_1, \mathrm{X}_2, R\})$ *has no perfect matching* **do**
8 $g \leftarrow \mathrm{argmin}_{h \in \mathrm{X}_2} v_1(h)$
9 $\mathrm{X}_2 \leftarrow \mathrm{X}_2 \setminus \{g\}$
10 $R \leftarrow R \cup \{g\}$
11 Update the edges of G_{EFx}
12 **else**
13 $(\mathrm{X}_2', \mathrm{X}_2'') \leftarrow$ leximin++$(\mathrm{X}_2, v_1^{\max}, v_1^{\max})$
14 Construct $G_{\text{EFx}}(\{1, 2\}, \{\mathrm{X}_1, \mathrm{X}_2', \mathrm{X}_2''\})$
/* The relevant case above yields G_{EFx} */
15 Let $\{(1, \mathrm{X}_1^*), (2, \mathrm{X}_2^*)\}$ be a perfect matching in G_{EFx} which
16 (a) maximizes the value of agent 2
17 (b) maximizes the value of agent 1 subject to (a)
18 **return** $(S_1^{\max}(\mathrm{X}_1^*), S_2^{\max}(\mathrm{X}_2^*))$

Note that every agent has at least one edge in G_{EFx} since they must be EFx-feasible with at least their favorite bundle.

Lemma 1. *If any agent has at least two edges in G_{EFx}, then G_{EFx} has a perfect matching.*

Proof. Note that every agent has at least one edge in G_{EFx} (an edge to the bundle they value the most). Therefore, if one of the two agents has two edges, we can safely let the other agent choose their favorite bundle first and then let the agent with the two agents choose second, ensuring that at least one of her two EFx-feasible bundles will still be available, leading to a perfect matching in G_{EFx}. □

Lemma 2. *If $v_1(\mathrm{X}_1) \geq \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$, Procedure 1 always finds a perfect matching in G_{EFx}.*

Proof. If the input allocation forms a perfect matching, we are done. Otherwise, both agents are initially only feasible with a single bundle, by Lemma 1, and since there is no perfect matching, both agents must only be feasible with the same bundle. By the reindexing in line 4, this bundle is X_2.

Removing items from X_2 will eventually create a new edge in G_{EFx} for agent 1, either towards X_1 or R. Since R is initially empty and agent 1 moves one item from X_2 to R at each iteration, there exists an iteration t such that $v_1^{\max}(\mathrm{X}_2^t) < \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$, where X_2 is the input bundle. Let t be the first such iteration. If a perfect matching is found before iteration $t-1$, we are done. Otherwise, in iteration $t-1$, we have $v_1^{\max}(\mathrm{X}_2^{t-1}) \geq \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$. Since $v_1^{\max}(\mathrm{X}_2^t) = v_1^{\max}(\mathrm{X}_2^{t-1} \setminus \{g_t\}) < \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$, it must be that in iteration $t-1$, agent 1 becomes EFx-feasible with X_1 due to the assumption that $v_1(\mathrm{X}_1) \geq \frac{1}{2} v_1^{\max}(\mathrm{X}_2)$.

As shown earlier in this proof, agent is initially EFx-feasible with only X_2. By Lemma 1, since a perfect matching was not found before iteration $t-1$, it must be that agent 1 had a single feasibility edge throughout these iterations, and so at each iteration $t' < t-1$, it must be that $v_1^{\max}(\mathrm{X}_2^{t'} \setminus \{g_{t'}\}) = v_1^{\max}(\mathrm{X}_2^{t'+1}) > v_1(\mathrm{X}_1)$ (note that g

is picked in such a way that this inequality holds). Thus, in iteration $t-1$, agent 1 is also EFx-feasible with X_2^{t-1}.

Since agent 1 is EFx-feasible with both X_1 and X_2^{t-1} in iteration $t-1$, a perfect matching exists by Lemma 1. □

Lemma 3. *If $v_1(X_1) \geq \frac{1}{2}v_1^{\max}(X_2)$, Procedure 1 returns an EFx allocation such that agent 1 gets a weakly higher value than her initial value, while agent 2 gets at least half of her initial value, Moreover, no agent envies the unallocated bundle.*

Proof. Note that a perfect matching in G_{EFx} implies an EFx allocation. If the input allocation forms a perfect matching, we are done. Otherwise, by Lemma 2, a perfect matching always exists. Let $t-1$ be the final iteration of the while loop (before a matching is returned), and X_1^* and X_2^* be the bundles in the returned matching assigned to agents 1 and 2. Note that $\max\left\{v_2(X_2^{t-1}), v_2(R^{t-1})\right\} \geq \frac{1}{2}v_2(X_2)$, where X_2 is the input bundle, since $X_2^{t-1} \cup R^{t-1} = X_2$. Since agent 2 is EFx-feasible with her favorite bundle and the returned matching maximizes agent 2's value, it must be that

$$v_2^{\max}(\mathbf{X}_2^*) \geq \max\left\{v_2(X_1), v_2(X_2^{t-1}), v_2(R^{t-1})\right\} \geq \frac{1}{2}v_2(X_2).$$

Also, agent 2 does not envy the unallocated bundle, since $\mathbf{X}_2^*$ is her favorite bundle.

By the argument of Lemma 2, agent 1 becomes EFx-feasible with X_1 in iteration $t'-1$, where t' is the earliest iteration for which $v_1^{\max}(X_2^{t'}) < \frac{1}{2}v_1^{\max}(X_2)$. Notice that $v_1^{\max}(X_2^{t'-1}) > v_1^{\max}(R^{t'-1})$, since $X_2^{t'-1} \cup R^{t'-1} = X_2$. Therefore, agent 1 could not have been EFx-feasible with $R^{t'-2}$ in iteration $t'-2$. This implies that in the returned matching, agent 1 can be matched to either X_1 or $X_2^{t'-1} = X_2^{t-1}$. If she is matched to X_1, then clearly $v_1^{\max}(X_1^*) = v_1(X_1)$, and if she is matched to X_2, then $v_1^{\max}(X_1^*) = v_1^{\max}(X_2^{t-1}) > v_1(X_1)$ by the argument of Lemma 2. Furthermore, agent 1 will similarly not envy the unallocated bundle since otherwise matching her to this bundle would increase her value and violate condition (b) in line 17. □

Lemma 4. *If $v_1(X_1) < \frac{1}{2}v_1^{\max}(X_2)$, Procedure 1 always finds a perfect matching in G_{EFx}.*

Proof. If the input allocation forms a perfect matching, we are done. Otherwise, since $X_2' \cup X_2'' = X_2$, it must be that $\max\{v_1^{\max}(X_2'), v_1^{\max}(X_2'')\} \geq \frac{1}{2}v_1^{\max}(X_2) > v_1(X_1)$. Since one of X_2' and X_2'' is agent 1's favorite bundle (wlog X_2'), and so it must be EFx-feasible. Note that $v_1^{\max}(X_2'') \geq v_2^{\max}(X_2' \setminus \{g\})$ for any $g \in X_2'$ by the outcome of `leximin++`. If $v_1(X_1) < v_1^{\max}(X_2'')$, then agent 1 is also EFx-feasible with $v_1^{\max}(X_2'')$. Otherwise, if $v_1(X_1) \geq v_1^{\max}(X_2'')$, then $v_1(X_1) \geq v_2^{\max}(X_2' \setminus \{g\})$ and so agent 1 is also EFx-feasible with X_1. Since agent 1 is EFx-feasible with at least two bundles, a perfect matching exists. □

Lemma 5. *If $v_1(X_1) < \frac{1}{2}v_1^{\max}(X_2)$, Procedure 1 returns an EFx allocation such that one agent gets a weakly higher value than her initial value, while the other gets at least half of her initial value, Moreover, no agent envies the unallocated bundle.*

Proof. Lemma 4 implies the existence of a perfect matching and also that agent 1 is EFx-feasible with at least two bundles. Note that $\max\{v_2(X_2'), v_2(X_2'')\} \geq \frac{1}{2}v_2(X_2)$, since $X_2' \cup X_2'' = X_2$. Since agent 2 is EFx-feasible with her favorite bundle and the returned matching maximizes agent 2's value, it must be that

$$v_2^{\max}(X_2^*) \geq \max\left\{v_2(X_2'), v_2(X_2'')\right\} \geq \frac{1}{2}v_2(X_2).$$

Also, for the same reason, she will not envy the unallocated bundle.

Lemma 4 also implies that at least two of agent 1's feasible bundles will have value at least $v_1(X_1)$ for her. Since the returned matching will allocate at least one of these two bundles to agent 1 (by the maximality of condition (b) in line 17) and so her value $v_1^{\max}(X_1^*)$ for her allocated bundle is weakly better than her initial value. Moreover, agent 1 will not envy the unallocated bundle or otherwise, we could increase agent 1's value and contradict the maximality of condition (b) in line 17. □

We now show the main result of this section.

Theorem 2. *`EFx-2A` returns a budget-feasible EFx allocation such that one agent gets a weakly higher value than her initial value, while the other gets at least half of her initial value, Moreover, no agent envies the unallocated bundle.*

Proof. Note that the returned allocation is budget-feasible by definition of $S^{\max}$. Combining Lemmas 3 and 5 gives the desired efficiency and envy guarantees. □

Note that since `EFx-2A` returns a budget-feasible allocation, some items from the matched bundles, X_1^* and X_2^*, might be left out (if the agent they are allocated to does not have sufficient budget for the whole bundle). In Lemma 6, we show that this is the case with at most one of X_1^* and X_2^*, and the left-out part is not EFx-envied (respectively envied) by the agents. We use this lemma for the three agents procedure later on.

Lemma 6. *In the allocation returned by `EFx-2A`, one of X_1^* and X_2^* will be completely allocated (to the agent with the higher budget), while the other bundle may be partially allocated. If R' is the left-out part, then one agent will not envy R', while the other will not EFx-envy it.*

4 EFx Allocations for 3-Agent Instances

In the case of two-agent instances, we were able to design a somewhat simple procedure in which the envying agent splits the envied bundle into two parts in such a way that an EFx allocation with optimal efficiency is returned. In the case of three-agent instances, however, things become significantly more complicated, since two agents may envy the third agent, yet disagree on how their bundle should be split. Furthermore, the fact that each of the three agents can have a vastly different budget, adds to the complexity of the problem. To address this issue, we first try to reduce the problem to the case of equal budgets. In order to do so, we consider the "monopoly value" of the higher-budget agents, if we were to limit their budget to the lowest one (see Section 2 for a definition). If all agents' monopoly value with the smallest budget is "high enough" (this threshold is determined by a parameter α whose exact value we determine later on), then we reduce the problem to the case of equal budgets. If not, then we essentially let the agents arrive in increasing order of their budgets and choose their preferred budget-feasible bundle among the remaining goods. Note that we assume that $|M| > 3$, since otherwise, the problem is trivial - the maximum `NSW` allocation is EFx.

Our main procedure, Procedure 2, first reindexes the agents in increasing order of their budgets, such that agent 1 has the smallest budget, followed by agent 2, and then 3. Then, it runs a preprocessing phase (described in Section 4.1) which allows each agent to set aside a single "high value" good. Next, our procedure checks whether we can reduce the input instance to one where all agents have equal budgets B_1, while still guaranteeing constant efficiency

through the monopoly value $m(B_1)$, and if so, we call Procedure 4 which returns an EFx allocation for instances with equal budgets (see Section 4.2). Otherwise, we run Procedure 5 (see Section 4.3). Once one of these subroutines returns an allocation, we provide each agent with the option of dropping the bundle they were assigned in exchange for the "high-value" good set aside for them during the pre-processing phase. This ensures that agents with a lot of value concentrated on a single item have the opportunity to achieve at least that much value if it is set aside for them in the pre-processing phase.

Procedure 2: (EFx-3A) EFx allocation for 3 agents

Input: Parameter α
Reindex the agents in increasing order of their budgets, such that $B_1 \leq B_2 \leq B_3$
$M, (s_1, s_2, s_3) \leftarrow$ Procedure 3
if $m_2(B_1) \geq \alpha$ *and* $m_3(B_1) \geq \alpha$ **then**
    $B_2, B_3 \leftarrow B_1$
    $\mathbf{X} \leftarrow$ Procedure 4
else
    $\mathbf{X} \leftarrow$ Procedure 5
for $i \in N$ **do**
    if $v_i(s_i) > v_i(\mathrm{X}_i)$ **then**
        $\mathrm{X}_i \leftarrow s_i$
return $\mathbf{X}$

4.1 The Pre-Processing Step: Procedure 3

We now present the pre-processing subroutine that we rely on to set aside one item for each agent. This subroutine tries to match them with one of their favorite three items and set it aside to ensure that they will never end up with a lower value. We carefully design this procedure so that this matching satisfies some useful properties that we use later on to prove the desired EFx and efficiency guarantees. In more detail, we use the maximum NSW allocation to "prioritize" the agents who have a favorite item in their respective maximum NSW bundle. Specifically, we guarantee that all these agents will be matched to a good that is at least as valuable as the most valuable good in their bundle (note that this can be achieved by just matching each such agent with her favorite good in her bundle). Then, given this hard constraint, we choose the maximum weight matching using the agents' values for the corresponding goods as the weights. This procedure returns the goods matched to each agent (observe that each agent gets one), sets them aside, and removes them from M. Note that Procedure 2, at the very end, gives each agent the option to pick their set-aside item if that would increase their value, which acts as a safety net for their value guarantee.

From this point on, let $\mathbf{X}^{\mathrm{OPT}}$ be the maximum NSW allocation on the remaining items $M \setminus \{s_1, s_2, s_3\}$. We now lower bound each agent's value in the maximum NSW allocation on the remaining items using the following lemma.

Lemma 7. *At the end of Procedure 3, at least one of the following three cases must hold for agents i, j, k:*

- $v_i(\mathrm{X}_i^{OPT}) \geq 1 - 3v_i(s_i)$ *and* $v_j(\mathrm{X}_j^{OPT}) = v_k(\mathrm{X}_k^{OPT}) = 1$
- $v_i(\mathrm{X}_i^{OPT}) \geq 1 - 2v_i(s_i)$ *and* $v_j(\mathrm{X}_j^{OPT}) \geq 1 - v_j(s_j)$ *and* $v_k(\mathrm{X}_k^{OPT}) = 1$
- $v_i(\mathrm{X}_i^{OPT}) \geq 1 - v_i(s_i)$ *and* $v_j(\mathrm{X}_j^{OPT}) \geq 1 - v_j(s_j)$ *and* $v_k(\mathrm{X}_k^{OPT}) \geq 1 - v_k(s_k)$

Procedure 3: Pre-processing for 3 agents

Input: Set of items M, maximum NSW allocation $\mathbf{X}^{\mathrm{OPT}}(M)$
For each agent i, let S_i be the set of their most favorite three budget-feasible goods in M
Let G be a bipartite graph with vertices $\{1, 2, 3\}$, corresponding to agents, on one side, and vertices $S_1 \cup S_2 \cup S_3$, corresponding to goods, on the other. An edge with weight $v_i(g)$ connects each agent i to each good $g \in S_i$
Let $\mathcal{P}$ be the set of all matchings of G such that every agent i with $S_i^* = S_i \cap \mathrm{X}_i^{\mathrm{OPT}} \neq \emptyset$ is matched to an item of value at least $\max_{g \in S_i^*} v_i(g)$
Let P be a maximum weight matching in $\mathcal{P}$ and for each agent i, let s_i be the good they are matched to in P
$M \leftarrow M \setminus \{s_1, s_2, s_3\}$
return $M, (s_1, s_2, s_3)$

4.2 The Equal Budget Case: Procedure 4

We now introduce Procedure 4 which outputs an EFx allocation with high NSW when the agents have equal budgets. The main idea behind this procedure is to reduce the budget-feasible problem to the case of unrestricted additive valuations. To do this, it removes a fraction of each agent's bundle in the NSW maximizing allocation, so that the set of all remaining items Z is affordable for every agent, i.e., the total cost across these items is within the common budget B_1. It does so, however, while keeping enough value for every agent. The procedure then computes a complete EFx allocation (i.e., all items in Z are allocated), which was proven to exist by Chaudhury et al. [10] - note that their existence proof is algorithmic, so we can use this approach to find a complete EFx allocation. Lastly, as an optimization, we allow agents to swap bundles with each other if they envy each other since the computed EFx allocation only guarantees the absence of EFx-envy. Note that before the procedure starts, we assume that the budgets are normalized to 1, which is without loss of generality.

We crucially use the fact that the value of every agent for any item is upper bounded by the value of their set-aside item to ensure that they can maintain roughly $1/n$ of their original value using at most $1/n$ of their total budget. Note that our procedure can handle any number of agents, however, we use $n = 3$ for most of the analysis.

Procedure 4: EFx allocation for 3 agents with equal budgets

Input: Maximum NSW allocation $\mathbf{X} = \mathbf{X}^{\mathrm{OPT}}(M)$
for $i \in N$ **do**
    while $c(\mathrm{X}_i) > \frac{1}{n}$ **do**
        $\mathrm{X}_i \leftarrow \mathrm{X}_i \setminus \left\{ \operatorname{argmin}_{g \in \mathrm{X}_i} \frac{v_i(g)}{c(g)} \right\}$
$Z \leftarrow \cup_{i \in N} \mathrm{X}_i$
$\mathbf{X}^{\mathrm{ALG}} \leftarrow$ a complete EFx allocation of Z
if *i envies j and j envies k and k envies i for any agents i, j, k* **then**
    $\mathrm{X}_i^{\mathrm{ALG}}, \mathrm{X}_j^{\mathrm{ALG}}, \mathrm{X}_k^{\mathrm{ALG}} \leftarrow \mathrm{X}_j^{\mathrm{ALG}}, \mathrm{X}_k^{\mathrm{ALG}}, \mathrm{X}_i^{\mathrm{ALG}}$
while *i envies j for any agents i, j* **do**
    $\mathrm{X}_i^{\mathrm{ALG}}, \mathrm{X}_j^{\mathrm{ALG}} \leftarrow \mathrm{X}_j^{\mathrm{ALG}}, \mathrm{X}_i^{\mathrm{ALG}}$
return $\mathbf{X}^{\mathrm{ALG}}$

Lemma 8. *After line 4 of Procedure 4, $v_i(\mathrm{X}_i) \geq \frac{v_i(\mathrm{X}_i^{OPT})}{n} - v_i(s_i)$ for all agents $i \in \{1, \ldots, n\}$.*

Lemma 9. *For three-agent instances with equal budgets, Procedure 4 returns a budget-feasible EFx allocation with the following individual guarantees:*

- $v_i(\mathrm{X}_i^{ALG}) \geq \frac{v_i(\mathrm{X}_i^{OPT})}{9} - \frac{v_i(s_i)}{3}$ *for an agent* i
- $v_j(\mathrm{X}_j^{ALG}) \geq \frac{v_j(\mathrm{X}_j^{OPT})}{9} - \frac{2v_j(s_j)}{3}$ *for an agent* $j \neq i$
- $v_k(\mathrm{X}_k^{ALG}) \geq \frac{v_k(\mathrm{X}_k^{OPT})}{9} - v_k(s_k)$ *for an agent* $k \neq i, j$

Lemma 10. *Procedure 4 returns an allocation* $\mathbf{X}^{ALG}$ *that is budget-feasible, EFx, and* $\sqrt[3]{\frac{\alpha^2}{15 \cdot 15 \cdot 18}}$*-efficient.*

4.3 The Remaining Case: Procedure 5

Below, we present the description of Procedure 5, which we call when some agent cannot be satisfied with a reduced budget. This procedure starts by assigning agent 1 her monopoly bundle and running EFx-2A on agents 2 and 3 with the remaining items. Intuitively, the subroutine checks how much more value agents 2 and 3 have for agent 1's bundle, handling the cases where at least one of them would lose a lot of value if their budget was reduced to B_1. If agents 2 and 3 do not envy agent 1, then we return the allocation. The case when one agent envies agent 1 and the other is "far" away from envying agent 1 (that will be agent 3 or otherwise we reindex the agents without loss) is more tricky and requires more careful analysis. At a high level, since agent 3 has much more value for her bundle compared to any other bundle she could get with a budget of B_1, she can afford to lose a few subsets of her bundle that cost at most B_1. Note that both agents 1 and 2 split agent 1's monopoly bundle, which fits into a budget of B_1, so as long as we guarantee that agent 3 gets a bundle that she values more than anything she could get with a budget of B_1, she will not envy the other agents.

In order to guarantee that the final allocation is EFx, we use the round-robin procedure in line 11 on X_3 (agent 3's bundle). Intuitively, this procedure splits X_3 into two parts, in a round-robin fashion, such that the resulting parts, X_3' and X_3'', are "roughly" equal in value (up to one item) for agent 3.

Lemma 11. *Procedure 5 always returns an allocation that is budget-feasible, EFx for any* $\alpha \leq \frac{1}{35}$*, and* $\sqrt[3]{\frac{1-9\alpha}{6 \cdot 13 \cdot 21}}$*-efficient.*

We now present the main result of our paper, which combines the properties of Procedures 3, 4, and 5 to achieve an EFx allocation with constant NSW for three-agent instances.

Theorem 3. *For instances with 3 agents, Procedure EFx-3A returns a budget-feasible EFx allocation with constant efficiency.*

Procedure 5: Subroutine for Procedure 2

1 **Input**: Set of items M
2 $\mathrm{X}_1 \leftarrow S_1^{\max}(M)$
3 $(\bar{\mathrm{X}}_2, \bar{\mathrm{X}}_3) \leftarrow$ max NSW allocation for agents 2, 3 and items $M \setminus \mathrm{X}_1$
4 $(\mathrm{X}_2, \mathrm{X}_3) \leftarrow$ EFx-2A$(\{2,3\}, (\bar{\mathrm{X}}_2, \bar{\mathrm{X}}_3))$
5 **if** $m_2(B_1) < \alpha$ *and* $m_3(B_1) < \alpha$ **then**
6 **return** $(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3)$
7 Reindex agents 2 and 3 such that $m_3(B_1) < \alpha$
8 **if** $v_2(\mathrm{X}_2) \geq v_2(\mathrm{X}_1)$ **then**
9 **return** $(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3)$
10 $(\mathrm{X}_1', \mathrm{X}_2') \leftarrow$ EFx-2A$(\{1,2\}, (\emptyset, \mathrm{X}_1))$
11 Let agent 3 run round-robin by value with himself on X_3 and $(\mathrm{X}_3', \mathrm{X}_3'')$ be the two resulting partitions
12 $\mathrm{X}_3^* \leftarrow \mathrm{X}_3 \setminus \operatorname{argmax}_{S \in \{\mathrm{X}_3', \mathrm{X}_3''\}} v_2(S)$
13 $\mathrm{X}_1'' \leftarrow S_1^{\max}(\mathrm{X}_3^*)$
14 $\mathrm{X}_1^* \leftarrow \mathrm{X}_1'$
15 **if** $v_1(\mathrm{X}_1') < v_1(\mathrm{X}_1'')$ **then**
16 $\mathrm{X}_1^* \leftarrow \mathrm{X}_1''$
17 $\mathrm{X}_3^* \leftarrow \mathrm{X}_3^* \setminus \mathrm{X}_1''$
18 **return** $(\mathrm{X}_1^*, \mathrm{X}_2', \mathrm{X}_3^*)$

5 Conclusion and Open Problems

The main open problem is whether a constant approximation of the maximum Nash welfare is achievable through EFx budget-feasible allocations for an arbitrary number of agents. In fact, this question is open even for EF1 budget-feasible allocations. Note that in very recent and independent work, Barman et al. [5] extended the result of Chaudhury et al. [12] to allow for generalized assignment constraints. This captures our setting and implies the existence of budget-feasible EFx allocations in which no agent envies the charity for an arbitrary number of agents. While their algorithm does not yield a bounded approximation of the maximum NSW, we believe that a modification using some of the ideas introduced in our paper can be shown to achieve a constant approximation of the NSW objective for a constant number of agents and, in general, a $O(n)$ approximation. Specifically, in the modified algorithm each agent initially sets aside the most valuable item from the bundle they receive in the maximum NSW allocation (if such bundle is not empty) and then a budget-feasible EFx allocation (with respect to the charity as well) is computed on the remaining goods using the ComputeFEFx procedure described in [5]. Lastly, each agent can pick their preferred set of goods between the bundle allocated to them by ComputeFEFx and the single good that they set aside in the first step. Similarly to our analysis in this paper, setting aside an good for each agent acts as a safety net against situations in which a very valuable good for one agent is allocated to another agent or to charity, which would still achieve EFx but can severely reduce efficiency. However, even if this approach could guarantee a linear approximation to the maximum NSW objective, it does not look like it could be used to achieve a constant approximation, which (or better than $O(n)$), which remains an interesting open problem.

Another interesting direction is to study the approximation achievable via budget-feasible envy-free allocations of *divisible* goods. Although maximizing the NSW with divisible goods is always envy-free for (unrestricted) additive valuations, this is not true in the presence of budget constraints. For example, consider the following instance with two agents, both with a budget of 1, and two goods with cost $c(1) = c(2) = 1$, where the valuations are $v_1(1) = 1, v_1(2) = 0$, and $v_2(1) = 0.6, v_2(2) = 0.4$. The maximum NSW allocation gives agent 1 the whole good 1 and agent 2 the whole good 2 for a NSW of $\sqrt{0.4}$. However, the best envy-free allocation in terms of NSW gives agent 1 a 3/4 fraction of good 1 and agent 2 a 1/4 fraction of good 1 and a 3/4 fraction of good 2. This allocation has a NSW of $\sqrt{0.3375}$. Therefore, this instance yields no better than a 0.918 approximation of the maximum Nash welfare. What is the best approximation of NSW achievable using envy-free budget-feasible allocations of divisible goods?

Acknowledgements

The authors were partially supported by NSF CAREER award CCF 2047907.

References

[1] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris, 'Maximum Nash welfare and other stories about EFX', *Theor. Comput. Sci.*, **863**, 69–85, (2021).

[2] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris, 'Fair division of indivisible goods: A survey', in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 5385–5393, (2022).

[3] Haris Aziz, Bo Li, Hervé Moulin, and Xiaowei Wu, 'Algorithmic fair allocation of indivisible items: a survey and new questions', *SIGecom Exch.*, **20**(1), 24–40, (2022).

[4] Siddharth Barman, Arindam Khan, Sudarshan Shyam, and K. V. N. Sreenivas, 'Finding fair allocations under budget constraints', *CoRR*, **abs/2208.08168**, (2022).

[5] Siddharth Barman, Arindam Khan, Sudarshan Shyam, and K. V. N. Sreenivas, 'Guaranteeing envy-freeness under generalized assignment constraints', in *Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023, London, United Kingdom, July 9-12, 2023*, pp. 242–269. ACM, (2023).

[6] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat, 'Almost full EFX exists for four agents', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4826–4833, (2022).

[7] Eric Budish, 'The combinatorial assignment problem: approximate competitive equilibrium from equal incomes', *Journal of Political Economy*, **119**(6), 1061–1103, (2011).

[8] Ioannis Caragiannis, Nick Gravin, and Xin Huang, 'Envy-freeness up to any item with high Nash welfare: The virtue of donating items', in *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019*, pp. 527–545, (2019).

[9] 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).

[10] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn, 'EFX exists for three agents', in *Proceedings of the 21st ACM Conference on Economics and Computation*, pp. 1–19, (2020).

[11] Bhaskar Ray Chaudhury, Jugal Garg, and Ruta Mehta, 'Fair and efficient allocations under subadditive valuations', in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 5269–5276, (2021).

[12] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa, 'A little charity guarantees almost envy-freeness', *SIAM J. Comput.*, **50**(4), 1336–1358, (2021).

[13] Sijia Dai, Guichen Gao, Shengxin Liu, Boon Han Lim, Li Ning, Yicheng Xu, and Yong Zhang, 'EFX under budget constraint', in *Frontiers of Algorithmic Wisdom*, volume 13461, pp. 3–14, (2022).

[14] Michal Feldman, Simon Mauras, and Tomasz Ponitka, 'On optimal tradeoffs between EFX and Nash welfare', *CoRR*, **abs/2302.09633**, (2023).

[15] Jiarui Gan, Bo Li, and Xiaowei Wu, 'Approximately envy-free budget-feasible allocation', *CoRR*, **abs/2106.14446**, (2021).

[16] Marius Garbea, Vasilis Gkatzelis, and Xizhi Tan, 'Efx budget-feasible allocations with high nash welfare', *CoRR*, **abs/2305.02280**, (2023).

[17] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi, 'On approximately fair allocations of indivisible goods', in *Proceedings 5th ACM Conference on Electronic Commerce (EC-2004)*, pp. 125–131, (2004).

[18] Benjamin Plaut and Tim Roughgarden, 'Almost envy-freeness with general valuations', *SIAM J. Discret. Math.*, **34**(2), 1039–1068, (2020).

[19] Xiaowei Wu, Bo Li, and Jiarui Gan, 'Budget-feasible maximum Nash social welfare is almost envy-free', in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 465–471, (2021).